

GigaDevice Semiconductor Inc.

GD32W51x
Arm® Cortex®-M33 32-bit MCU

User Manual

Revision 1.0

(Nov 2021)

Table of Contents

Table of Contents.....	2
List of Figures	24
List of Tables	32
1. System and memory architecture	37
1.1. Arm Cortex-M33 processor.....	37
1.2. System architecture	38
1.3. TrustZone® overview.....	40
1.3.1. TrustZone® security attribution.....	40
1.3.2. Peripheral classification.....	42
1.4. Memory map.....	43
1.4.1. On-chip SRAM memory	47
1.4.2. SRAM1 Write protection.....	47
1.4.3. SRAM1 security protection	48
1.4.4. SRAM1 Erase.....	48
1.4.5. On-chip flash memory overview.....	48
1.5. Boot configuration	48
1.6. System configuration controller (SYSCFG).....	51
1.6.1. SYSCFG main features.....	51
1.6.2. SYSCFG TrustZone security and privilege	51
1.6.3. SYSCFG registers.....	52
1.7. Device electronic signature	64
1.7.1. Memory density information.....	64
1.7.2. Unique device ID (96 bits)	65
2. Flash memory controller (FMC)	67
2.1. Overview.....	67
2.2. Characteristics	67
2.3. System overview.....	67
2.4. Function overview.....	68
2.4.1. Flash memory architecture	68
2.4.2. Read operations	69
2.4.3. Unlock the FMC_CTL/FMC_SECCTL register	70
2.4.4. Page erase	71
2.4.5. Mass erase.....	72
2.4.6. Main flash programming.....	74
2.4.7. Option bytes.....	75

2.4.8. TrustZone security protection.....	77
2.4.9. Security protection.....	78
2.4.10. Write protection.....	82
2.4.11. Flash privileged and unprivileged mode.....	82
2.4.12. FLASH interrupts	82
2.5. Register definition.....	84
2.5.1. Unlock key register (FMC_KEY).....	84
2.5.2. Option byte unlock key register (FMC_OBKEY).....	84
2.5.3. Status register (FMC_STAT).....	85
2.5.4. Control register (FMC_CTL).....	85
2.5.5. Address register (FMC_ADDR)	87
2.5.6. Option byte status register (FMC_OBSTAT)	88
2.5.7. Secure Unlock key register (FMC_SECKEY).....	88
2.5.8. Secure status register (FMC_SECSTAT).....	89
2.5.9. Secure Control register (FMC_SECCTL)	90
2.5.10. Secure Address register (FMC_SECADDR)	91
2.5.11. Option byte register (FMC_OBR).....	92
2.5.12. Option byte user register (FMC_OBUSER).....	93
2.5.13. Secure mark configuration register 0 (FMC_SECMCFG0).....	93
2.5.14. Secure dedicated mark protection register 0 (FMC_DMP0).....	94
2.5.15. Option byte write protection area register 0 (FMC_OBWRP0)	94
2.5.16. Secure mark configuration register 1 (FMC_SECMCFG1).....	95
2.5.17. Secure dedicated mark protection register 1 (FMC_DMP1).....	96
2.5.18. Option byte write protection area register 1 (FMC_OBWRP1)	96
2.5.19. Secure mark configuration register 2(FMC_SECMCFG2).....	97
2.5.20. Secure mark configuration register 3 (FMC_SECMCFG3).....	98
2.5.21. NO-RTDEC region register x (FMC_NODECx, x=0,1,2,3).....	98
2.5.22. Offset region register (FMC_OFRG).....	99
2.5.23. Offset value register (FMC_OFVR)	99
2.5.24. Secure dedicated mark protection control register (FMC_DMPCTL).....	100
2.5.25. Privilege configuration register (FMC_PRIVCFG).....	101
2.5.26. Product ID register (FMC_PID).....	101
3. Electronic fuse (EFUSE).....	102
3.1. Overview.....	102
3.2. Characteristics	102
3.3. Block diagram.....	102
3.4. Function overview.....	103
3.4.1. EFUSE architecture.....	103
3.4.2. EFUSE macro description	103
3.4.3. Read operation.....	107
3.4.4. Program operation.....	107

3.5. Register definition.....	108
3.5.1. Control and status register (EFUSE_CS)	108
3.5.2. Address register (EFUSE_ADDR).....	109
3.5.3. Control register (EFUSE_CTL).....	110
3.5.4. Trustzone control register(EFUSE_TZCTL)	111
3.5.5. Flash protection control register (EFUSE_FP_CTL).....	112
3.5.6. User control register (EFUSE_USER_CTL).....	112
3.5.7. MCU initialization data register (EFUSE MCU_INIT_DATA).....	113
3.5.8. Firmware AES key register (EFUSE_AES_KEY).....	114
3.5.9. RoTPK key register (EFUSE_ROTAK_KEY)	114
3.5.10. Debug password register (EFUSE_DP).....	114
3.5.11. IAK key or GSSA register (EFUSE_IAK_GSSA).....	115
3.5.12. Product UID register (EFUSE_PUID)	115
3.5.13. HUK key register (EFUSE_HUK_KEY).....	116
3.5.14. RF data register (EFUSE_RF_DATA).....	116
3.5.15. User data register (EFUSE_USER_DATA).....	116
3.5.16. EFUSE Pre-TrustZone enable register (EFUSE_PRE_TZEN)	117
3.5.17. TrustZone boot address register (EFUSE_TZ_BOOT_ADDR).....	117
3.5.18. No-TrustZone boot address register (EFUSE_NTZ_BOOT_ADDR).....	118
4. Instruction cache (ICACHE).....	119
4.1. Introduction.....	119
4.2. Characteristics.....	119
4.3. Function overview.....	119
4.3.1. ICACHE initialization.....	120
4.3.2. Paired master cache.....	120
4.3.3. ICACHE TAG memory	121
4.3.4. Address remapping	121
4.3.5. Cacheable and uncacheable access	123
4.3.6. ICACHE performance monitoring.....	124
4.3.7. ICACHE interrupts.....	124
4.4. Register definition.....	126
4.4.1. Control register (ICACHE_CTL).....	126
4.4.2. Status register (ICACHE_STAT)	127
4.4.3. Interrupt enable register (ICACHE_INTEN)	127
4.4.4. Flag clear register (ICACHE_FC).....	128
4.4.5. Hit monitor counter register (ICACHE_HMC).....	128
4.4.6. Miss monitor counter register (ICACHE_MMC).....	129
4.4.7. Configuration register (ICACHE_CFGx).....	129
5. Power management unit (PMU)	131
5.1. Overview.....	131
5.2. Characteristics.....	131

5.3. Function overview.....	132
5.3.1. Battery backup domain	132
5.3.2. VDD / VDDA power domain	133
5.3.3. 1.2V power domain	135
5.3.4. Power saving modes	136
5.3.5. Security description.....	141
5.4. Register definition.....	143
5.4.1. Control register 0 (PMU_CTL0)	143
5.4.2. Control and status register 0 (PMU_CS0).....	145
5.4.3. Control register 1 (PMU_CTL1)	147
5.4.4. Control and status register 1 (PMU_CS1).....	148
5.4.5. RF Control register (PMU_RFCTL)	148
5.4.6. Secure configuration register (PMU_SECCFG).....	149
5.4.7. Privilege configuration register (PMU_PRICFG).....	150
6. Reset and clock unit (RCU)	152
6.1. Reset control unit (RCTL)	152
6.1.1. Overview	152
6.1.2. Function overview	152
6.2. Clock control unit (CCTL)	153
6.2.1. Overview	153
6.2.2. Characteristics.....	155
6.2.3. Function overview	156
6.3. RCU security protection.....	159
6.4. RCU privilege protection.....	161
6.5. Register definition.....	163
6.5.1. Control register (RCU_CTL).....	163
6.5.2. PLL register (RCU_PLL).....	166
6.5.3. Clock configuration register 0 (RCU_CFG0).....	167
6.5.4. Clock interrupt register (RCU_INT)	170
6.5.5. AHB1 reset register (RCU_AHB1RST)	173
6.5.6. AHB2 reset register (RCU_AHB2RST)	174
6.5.7. AHB3 reset register (RCU_AHB3RST)	175
6.5.8. APB1 reset register (RCU_APB1RST).....	176
6.5.9. APB2 reset register (RCU_APB2RST).....	178
6.5.10. AHB1 enable register (RCU_AHB1EN).....	179
6.5.11. AHB2 enable register (RCU_AHB2EN).....	182
6.5.12. AHB3 enable register (RCU_AHB3EN).....	183
6.5.13. APB1 enable register (RCU_APB1EN).....	183
6.5.14. APB2 enable register (RCU_APB2EN).....	185
6.5.15. AHB1 sleep mode enable register (RCU_AHB1SPEN).....	187
6.5.16. AHB2 sleep mode enable register (RCU_AHB2SPEN).....	189

6.5.17.	AHB3 sleep mode enable register (RCU_AHB3SPEN).....	190
6.5.18.	APB1 sleep mode enable register (RCU_APB1SPEN).....	190
6.5.19.	APB2 sleep mode enable register (RCU_APB2SPEN).....	192
6.5.20.	Backup domain control register (RCU_BDCTL).....	194
6.5.21.	Reset source/clock register (RCU_RSTSCK).....	195
6.5.22.	PLL clock spread spectrum control register (RCU_PLLSSCTL)	197
6.5.23.	PLL clock configuration register (RCU_PLLCFG)	198
6.5.24.	Clock configuration register 1 (RCU_CFG1).....	199
6.5.25.	Additional clock control register (RCU_ADDCTL)	202
6.5.26.	Secure protection configuration register (RCU_SECP_CFG).....	203
6.5.27.	Secure protection status register (RCU_SECP_STAT)	205
6.5.28.	AHB1 secure protection status register (RCU_AHB1SECP_STAT)	206
6.5.29.	AHB2 secure protection status register (RCU_AHB2SECP_STAT)	209
6.5.30.	AHB3 secure protection status register (RCU_AHB3SECP_STAT)	210
6.5.31.	APB1 secure protection status register (RCU_APB1SECP_STAT)	210
6.5.32.	APB2 secure protection status register (RCU_APB2SECP_STAT)	212
6.5.33.	Voltage key register (RCU_VKEY).....	214
6.5.34.	Deep-sleep mode voltage register (RCU_DSV).....	214
7.	Interrupt/event controller (EXTI).....	216
7.1.	Overview.....	216
7.2.	Characteristics.....	216
7.3.	Interrupts function overview.....	216
7.4.	External interrupt and event (EXTI) block diagram.....	220
7.5.	External interrupt and Event function overview.....	220
7.6.	EXTI event protection.....	222
7.7.	EXTI security protection	223
7.8.	EXTI privilege protection	223
7.9.	Register definition.....	224
7.9.1.	Interrupt enable register (EXTI_INTEN)	224
7.9.2.	Event enable register (EXTI_EVENT).....	224
7.9.3.	Rising edge trigger enable register (EXTI_RTEN).....	225
7.9.4.	Falling edge trigger enable register (EXTI_FTEN)	226
7.9.5.	Software interrupt event register (EXTI_SWIEV).....	226
7.9.6.	Pending register (EXTI_PD).....	227
7.9.7.	Security configuration register (EXTI_SECCFG)	228
7.9.8.	Privilege configuration register (EXTI_PRIVCFG).....	228
7.9.9.	Lock register (EXTI_LOCK)	229
8.	General-purpose and alternate-function I/Os (GPIO and AFIO).....	230
8.1.	Overview.....	230

8.2. Characteristics	230
8.3. Function overview.....	230
8.3.1. GPIO pin configuration.....	232
8.3.2. External interrupt/event lines.....	233
8.3.3. Alternate functions (AF).....	233
8.3.4. Additional functions	233
8.3.5. Input configuration.....	233
8.3.6. Output configuration.....	233
8.3.7. Analog configuration.....	234
8.3.8. Alternate function (AF) configuration	235
8.3.9. GPIO locking function.....	235
8.3.10. GPIO single cycle toggle function	236
8.4. TrustZone GPIO security.....	236
8.5. Register definition.....	237
8.5.1. Port control register (GPIO _x _CTL, x=A..C).....	237
8.5.2. Port output mode register (GPIO _x _OMODE, x=A..C).....	239
8.5.3. Port output speed register (GPIO _x _OSPD, x=A..C)	241
8.5.4. Port pull-up/pull-down register (GPIO _x _PUD, x=A..C).....	243
8.5.5. Port input status register (GPIO _x _ISTAT, x=A..C).....	245
8.5.6. Port output control register (GPIO _x _OCTL, x=A..C).....	245
8.5.7. Port bit operate register (GPIO _x _BOP, x=A..C)	246
8.5.8. Port configuration lock register (GPIO _x _LOCK, x=A..C)	246
8.5.9. Alternate function selected register 0 (GPIO _x _AFSEL0, x=A..C).....	247
8.5.10. Alternate function selected register 1 (GPIO _x _AFSEL1, x=A..C).....	248
8.5.11. Bit clear register (GPIO _x _BC, x=A..C).....	250
8.5.12. Port bit toggle register (GPIO _x _TG, x=A..C).....	250
8.5.13. GPIO secure configuration register (GPIO _x _SCFG) (x=A..C).....	251
9. TrustZone protection controller union (TZPCU).....	252
9.1. Overview.....	252
9.2. Characteristics	252
9.3. Function overview.....	252
9.3.1. Block diagram.....	252
9.3.2. Illegal access definition.....	254
9.3.3. TrustZone® security privilege controller (TZSPC).....	255
9.3.4. TrustZone® block based memory protection controller (TzbMpc).....	256
9.3.5. TrustZone® illegal access controller (Tziac).....	256
9.3.6. SPC/GSSA debug	257
9.4. TZSPC Register definition	258
9.4.1. TZSPC control register (TZPCU_TZSPC_CTL).....	258
9.4.2. TZSPC secure access mode configuration register 0 (TZPCU_TZSPC_SAM_CFG0).....	258
9.4.3. TZSPC secure access mode configuration register 1 (TZPCU_TZSPC_SAM_CFG1).....	261

9.4.4.	TZSPC secure access mode configuration register 2 (TZPCU_TZSPC_SAM_CFG2).....	263
9.4.5.	TZSPC privilege access mode configuration register 0 (TZPCU_TZSPC_PAM_CFG0)....	264
9.4.6.	TZSPC privilege access mode configuration register 1 (TZPCU_TZSPC_PAM_CFG1)....	266
9.4.7.	TZSPC privilege access mode configuration register 2 (TZPCU_TZSPC_PAM_CFG2)....	269
9.4.8.	TZSPC external memory x non-secure mark register 0 (TZPCU_TZSPC_TZMMPCx_NS _M 0).....	270
9.4.9.	TZSPC external memory x non-secure mark register 1 (TZPCU_TZSPC_TZMMPCx_NS _M 1).....	271
9.4.10.	TZSPC external memory x non-secure mark register 2 (TZPCU_TZSPC_TZMMPCx_NS _M 2).....	272
9.4.11.	TZSPC external memory x non-secure mark register 3 (TZPCU_TZSPC_TZMMPCx_NS _M 3).....	273
9.4.12.	TZSPC debug configuration register (TZPCU_TZSPC_DBG_CFG)	274
9.5.	TZBMP_C0 registers definition.....	275
9.5.1.	TZBMP _C 0 control register (TZPCU_TZBMP _C 0_CTL).....	275
9.5.2.	TZBMP _C 0 vector register y (TZPCU_TZBMP _C 0_VEC _y)	276
9.5.3.	TZBMP _C 0 lock register 0 (TZPCU_TZBMP _C 0_LOCK0).....	276
9.6.	TZBMP_C1 registers definition.....	277
9.6.1.	TZBMP _C 1 control register (TZPCU_TZBMP _C 1_CTL).....	277
9.6.2.	TZBMP _C 1 vector register y (TZPCU_TZBMP _C 1_VEC _y)	278
9.6.3.	TZBMP _C 1 lock register 0 (TZPCU_TZBMP _C 1_LOCK0).....	279
9.7.	TZBMP_C2 registers definition.....	279
9.7.1.	TZBMP _C 2 control register (TZPCU_TZBMP _C 2_CTL).....	279
9.7.2.	TZBMP _C 2 vector register y (TZPCU_TZBMP _C 2_VEC _y)	280
9.7.3.	TZBMP _C 2 lock register 0 (TZPCU_TZBMP _C 2_LOCK0).....	281
9.8.	TZBMP_C3 registers definition.....	281
9.8.1.	TZBMP _C 3 control register (TZPCU_TZBMP _C 3_CTL).....	282
9.8.2.	TZBMP _C 3 vector register y (TZPCU_TZBMP _C 3_VEC _y)	282
9.8.3.	TZBMP _C 3 lock register 0 (TZPCU_TZBMP _C 3_LOCK0).....	283
9.9.	TZIAC Register definition.....	284
9.9.1.	TZIAC interrupt enable register 0 (TZPCU_TZIAC_INTEN0)	284
9.9.2.	TZIAC interrupt enable register 1 (TZPCU_TZIAC_INTEN1)	286
9.9.3.	TZIAC interrupt enable register 2 (TZPCU_TZIAC_INTEN2)	289
9.9.4.	TZIAC status register 0 (TZPCU_TZIAC_STAT0)	291
9.9.5.	TZIAC status register 1 (TZPCU_TZIAC_STAT1)	293
9.9.6.	TZIAC status register 2 (TZPCU_TZIAC_STAT2)	296
9.9.7.	TZIAC flag clear register 0 (TZPCU_TZIAC_STATC0).....	298
9.9.8.	TZIAC flag clear register 1 (TZPCU_TZIAC_STATC1).....	300
9.9.9.	TZIAC flag clear register 2 (TZPCU_TZIAC_STATC2).....	303
10.	CRC calculation unit (CRC)	306
10.1.	Overview	306

10.2. Characteristics.....	306
10.3. Function overview	307
10.4. Register definition	308
10.4.1. Data register (CRC_DATA).....	308
10.4.2. Free data register (CRC_FDATA)	308
10.4.3. Control register (CRC_CTL).....	309
11. True random number generator (TRNG)	310
11.1. Overview	310
11.2. Characteristics.....	310
11.3. Function overview	310
11.3.1. Operation flow	311
11.3.2. Error flags.....	311
11.4. Register definition	312
11.4.1. Control register (TRNG_CTL).....	312
11.4.2. Status register (TRNG_STAT)	312
11.4.3. Data register (TRNG_DATA).....	313
12. Direct memory access controller (DMA)	315
12.1. Overview	315
12.2. Characteristics.....	315
12.3. Block diagram.....	316
12.4. Function overview	317
12.4.1. Secure & priviledge	318
12.4.2. Peripheral handshake.....	319
12.4.3. Data process.....	321
12.4.4. Address generation	327
12.4.5. Circular mode.....	327
12.4.6. Switch-buffer mode.....	328
12.4.7. Transfer flow controller.....	328
12.4.8. Transfer operation.....	329
12.4.9. Transfer finish.....	330
12.4.10. Channel configuration.....	331
12.5. Interrupts.....	332
12.5.1. Flag	333
12.5.2. Exception.....	334
12.5.3. Error	335
12.6. Register definition	337
12.6.1. Interrupt flag register 0 (DMA_INTF0).....	337
12.6.2. Interrupt flag register 1 (DMA_INTF1).....	338

12.6.3.	Interrupt flag clear register 0 (DMA_INTC0).....	339
12.6.4.	Interrupt flag clear register 1 (DMA_INTC1).....	339
12.6.5.	Channel x control register (DMA_CHxCTL)	340
12.6.6.	Channel x counter register (DMA_CHxCNT).....	344
12.6.7.	Channel x peripheral base address register (DMA_CHxPADDR)	344
12.6.8.	Channel x memory 0 base address register (DMA_CHxM0ADDR).....	345
12.6.9.	Channel x memory 1 base address register (DMA_CHxM1ADDR).....	346
12.6.10.	Channel x FIFO control register (DMA_CHxFCTL)	346
12.6.11.	Security status register (DMA_SSTAT)	347
12.6.12.	Security status clear register (DMA_SSC).....	348
12.6.13.	Channel x security control register (DMA_CHxSCTL)	349
13.	Debug (DBG).....	351
13.1.	Overview	351
13.2.	JTAG/SW function description.....	351
13.2.1.	Switch JTAG or SW interface.....	351
13.2.2.	Pin assignment.....	351
13.2.3.	JTAG daisy chained structure.....	352
13.2.4.	Debug reset	352
13.2.5.	JEDEC-106 ID code.....	352
13.3.	Debug hold function description.....	352
13.3.1.	Debug support for power saving mode	352
13.3.2.	Debug support for TIMER, I2C, WWDGT, FWDGT and RTC	353
13.4.	DBG registers.....	354
13.4.1.	ID code register (DBG_ID).....	354
13.4.2.	Control register 0 (DBG_CTL0)	354
13.4.3.	Control register 1 (DBG_CTL1)	355
13.4.4.	Control register 2 (DBG_CTL2)	357
14.	Analog to digital converter (ADC)	358
14.1.	Overview	358
14.2.	Characteristics.....	358
14.3.	Pins and internal signals.....	359
14.4.	Function overview	360
14.4.1.	ADC clock.....	360
14.4.2.	ADCON switch.....	360
14.4.3.	Regular and inserted channel groups	360
14.4.4.	Conversion modes	361
14.4.5.	Inserted channel management.....	365
14.4.6.	Analog watchdog	366
14.4.7.	Data alignment	366
14.4.8.	Programmable sampling time.....	367

14.4.9. External trigger	367
14.4.10. DMA request	369
14.4.11. Overflow detection	369
14.4.12. Temperature sensor, internal reference voltage VREFINT and external battery voltage VBAT	
369	
14.4.13. On-chip hardware oversampling	370
14.4.14. ADC interrupts	372
14.5. Register definition	373
14.5.1. Status register (ADC_STAT)	373
14.5.2. Control register 0 (ADC_CTL0)	374
14.5.3. Control register 1 (ADC_CTL1)	376
14.5.4. Sample time register 0 (ADC_SAMPT0)	378
14.5.5. Sample time register 1 (ADC_SAMPT1)	379
14.5.6. Inserted channel data offset register x (ADC_IOFFx) (x=0..3)	380
14.5.7. Watchdog high threshold register (ADC_WDHT)	380
14.5.8. Watchdog low threshold register (ADC_WDLT)	380
14.5.9. Regular sequence register 0 (ADC_RSQ0)	381
14.5.10. Regular sequence register 1 (ADC_RSQ1)	381
14.5.11. Regular sequence register 2 (ADC_RSQ2)	382
14.5.12. Inserted sequence register (ADC_ISQ)	383
14.5.13. Inserted data register x (ADC_IDATAx) (x= 0..3)	383
14.5.14. Regular data register (ADC_RDATA)	384
14.5.15. Oversampling control register (ADC_OVSAMPCTL)	384
14.5.16. Common control register (ADC_CCTL)	386
15. Watchdog timer (WDGT)	387
15.1. Free watchdog timer (FWDGT)	387
15.1.1. Overview	387
15.1.2. Characteristics	387
15.1.3. Function overview	387
15.1.4. Register definition	390
15.2. Window watchdog timer (WWDT)	393
15.2.1. Overview	393
15.2.2. Characteristics	393
15.2.3. Function overview	393
15.2.4. Register definition	396
16. Real time clock (RTC)	398
16.1. Overview	398
16.2. Characteristics	398
16.3. Function overview	399
16.3.1. Block diagram	399
16.3.2. Clock source and prescalers	400

16.3.3.	Shadow registers introduction	400
16.3.4.	Configurable and field maskable alarm	400
16.3.5.	Configurable periodic auto-wakeup counter.....	401
16.3.6.	RTC initialization and configuration	401
16.3.7.	Calendar reading	403
16.3.8.	Resetting the RTC.....	404
16.3.9.	RTC shift function.....	404
16.3.10.	RTC reference clock detection	405
16.3.11.	RTC coarse digital calibration.....	406
16.3.12.	RTC smooth digital calibration.....	406
16.3.13.	Time-stamp function.....	408
16.3.14.	Tamper detection	409
16.3.15.	Calibration clock output.....	410
16.3.16.	Alarm output.....	410
16.3.17.	RTC security protection.....	410
16.3.18.	RTC privilege protection.....	413
16.3.19.	RTC backup registers protection.....	416
16.3.20.	RTC power saving mode management.....	417
16.3.21.	RTC interrupts	417
16.4.	Register definition	419
16.4.1.	Time register (RTC_TIME).....	419
16.4.2.	Date register (RTC_DATE).....	419
16.4.3.	Control register (RTC_CTL).....	420
16.4.4.	Initialization control and status register (RTC_ICSR).....	423
16.4.5.	Prescaler register (RTC_PSC).....	424
16.4.6.	Wakeup timer register (RTC_WUT).....	425
16.4.7.	Coarse calibration register (RTC_COSC).....	425
16.4.8.	Alarm 0 time and date register (RTC_ALRM0TD).....	426
16.4.9.	Alarm 1 time and date register (RTC_ALRM1TD).....	427
16.4.10.	Write protection key register (RTC_WPK).....	429
16.4.11.	Sub second register (RTC_SS)	429
16.4.12.	Shift function control register (RTC_SHIFTCTL).....	429
16.4.13.	Time of time stamp register (RTC_TTS).....	430
16.4.14.	Date of time stamp register (RTC_DTS).....	431
16.4.15.	Sub second of time stamp register (RTC_SSTS).....	432
16.4.16.	High resolution frequency compensation register (RTC_HRFC).....	432
16.4.17.	Tamper register (RTC_TAMP).....	433
16.4.18.	Alarm 0 sub second register (RTC_ALRM0SS).....	435
16.4.19.	Alarm 1 sub second register (RTC_ALRM1SS).....	436
16.4.20.	Privilege protection mode control register (RTC_PPM_CTL)	437
16.4.21.	Secure protection mode control register (RTC_SPM_CTL)	439
16.4.22.	Status register (RTC_STAT).....	440
16.4.23.	Non-secure masked interrupt status register (RTC_NSMI_STAT)	441

16.4.24. Secure masked interrupt status register (RTC_SMI_STAT).....	442
16.4.25. Status flag clear register (RTC_STATC).....	443
16.4.26. Backup registers (RTC_BKPx) (x=0..19).....	444
17. Timer (TIMERx).....	446
17.1. Advanced timer (TIMERx, x=0).....	447
17.1.1. Overview	447
17.1.2. Characteristics.....	447
17.1.3. Block diagram.....	448
17.1.4. Function overview	448
17.1.5. TIMERx registers(x=0).....	477
17.2. General level0 timer (TIMERx, x=1, 2, 3, 4).....	505
17.2.1. Overview	505
17.2.2. Characteristics.....	505
17.2.3. Block diagram.....	505
17.2.4. Function overview	506
17.2.5. TIMERx registers(x=1, 2, 3, 4).....	523
17.3. General level4 timer (TIMERx, x=15,16).....	549
17.3.1. Overview	549
17.3.2. Characteristics.....	549
17.3.3. Block diagram.....	549
17.3.4. Function overview	550
17.3.5. TIMERx registers(x=15, 16).....	564
17.4. Basic timer (TIMERx, x=5).....	580
17.4.1. Overview	580
17.4.2. Characteristics.....	580
17.4.3. Block diagram.....	580
17.4.4. Function overview	580
17.4.5. TIMERx registers(x=5).....	584
18. Universal synchronous/asynchronous receiver /transmitter (USART).....	589
18.1. Overview	589
18.2. Characteristics.....	589
18.3. Function overview	591
18.3.1. USART frame format	591
18.3.2. Baud rate generation.....	592
18.3.3. USART transmitter.....	593
18.3.4. USART receiver	594
18.3.5. Use DMA for data buffer access	595
18.3.6. Hardware flow control	597
18.3.7. Multi-processor communication.....	598
18.3.8. LIN mode.....	599

18.3.9.	Synchronous mode	600
18.3.10.	IrDA SIR ENDEC mode.....	601
18.3.11.	Half-duplex communication mode.....	602
18.3.12.	Smartcard (ISO7816-3) mode.....	602
18.3.13.	Auto baudrate detection.....	604
18.3.14.	ModBus communication.....	605
18.3.15.	Receive FIFO.....	605
18.3.16.	Wakeup from Deep-sleep mode	606
18.3.17.	USART interrupts	606
18.4.	Register definition	608
18.4.1.	Control register 0 (USART_CTL0).....	608
18.4.2.	Control register 1 (USART_CTL1).....	610
18.4.3.	Control register 2 (USART_CTL2).....	613
18.4.4.	Baud rate generator register (USART_BAUD)	616
18.4.5.	Prescaler and guard time configuration register (USART_GP).....	616
18.4.6.	Receiver timeout register (USART_RT)	617
18.4.7.	Command register (USART_CMD)	618
18.4.8.	Status register (USART_STAT)	619
18.4.9.	Interrupt status clear register (USART_INTC)	623
18.4.10.	Receive data register (USART_RDATA)	624
18.4.11.	Transmit data register (USART_TDATA)	625
18.4.12.	USART coherence control register (USART_CHC)	625
18.4.13.	USART receive FIFO control and status register (USART_RFCS)	626
19.	Inter-integrated circuit interface (I2C).....	628
19.1.	Overview	628
19.2.	Characteristics.....	628
19.3.	Function overview	628
19.3.1.	Clock requirements	629
19.3.2.	I2C communication flow	630
19.3.3.	Noise filter.....	633
19.3.4.	I2C timings.....	633
19.3.5.	Software reset.....	635
19.3.6.	Data transfer	635
19.3.7.	I2C slave mode.....	638
19.3.8.	I2C master mode	643
19.3.9.	SMBus support.....	648
19.3.10.	SMBus mode.....	651
19.3.11.	Wakeup from Deep-sleep mode	653
19.3.12.	Use DMA for data transfer	653
19.3.13.	I2C error and interrupts	654
19.3.14.	I2C debug mode	654

19.4. Register definition	655
19.4.1. Control register 0 (I2C_CTL0)	655
19.4.2. Control register 1 (I2C_CTL1)	657
19.4.3. Slave address register 0 (I2C_SADDR0)	659
19.4.4. Slave address register 1 (I2C_SADDR1)	660
19.4.5. Timing register (I2C_TIMING)	661
19.4.6. Timeout register (I2C_TIMEOUT)	662
19.4.7. Status register (I2C_STAT)	663
19.4.8. Status clear register (I2C_STATC)	666
19.4.9. PEC register (I2C_PEC).....	667
19.4.10. Receive data register (I2C_RDATA).....	667
19.4.11. Transmit data register (I2C_TDATA).....	668
19.4.12. Control register 2 (I2C_CTL2)	668
20. Serial peripheral interface/Inter-IC sound (SPI/I2S).....	670
20.1. Overview	670
20.2. Characteristics.....	670
20.2.1. SPI characteristics	670
20.2.2. I2S characteristics.....	670
20.3. SPI block diagram.....	671
20.4. SPI signal description.....	671
20.4.1. Normal configuration (Not Quad-SPI Mode).....	671
20.4.2. Quad-SPI configuration.....	672
20.5. SPI function overview.....	672
20.5.1. SPI clock timing and data format	672
20.5.2. NSS function.....	673
20.5.3. SPI operating modes	674
20.5.4. DMA function.....	682
20.5.5. CRC function.....	682
20.6. SPI interrupts	683
20.6.1. Status flags	683
20.6.2. Error flags.....	683
20.7. I2S block diagram.....	684
20.8. I2S signal description	685
20.9. I2S function overview	685
20.9.1. I2S audio standards.....	685
20.9.2. I2S clock.....	693
20.9.3. Operation.....	694
20.9.4. DMA function.....	697
20.10. I2S interrupts	698
20.10.1. Status flags	698

20.10.2. Error flags.....	698
20.11. Register definition	700
20.11.1. Control register 0 (SPI_CTL0)	700
20.11.2. Control register 1 (SPI_CTL1)	702
20.11.3. Status register (SPI_STAT).....	703
20.11.4. Data register (SPI_DATA).....	704
20.11.5. CRC polynomial register (SPI_CRCPOLY).....	705
20.11.6. RX CRC register (SPI_RCRC)	705
20.11.7. TX CRC register (SPI_TCRC)	706
20.11.8. I2S control register (SPI_I2SCTL).....	707
20.11.9. I2S clock prescaler register (SPI_I2SPSC).....	708
20.11.10. Quad-SPI mode control register (SPI_QCTL) of SPI0	709
21. Serial/Quad Parallel Interface (SQPI)	711
21.1. Overview	711
21.2. Characteristics.....	711
21.3. Function overview	711
21.3.1. SQPI mode definition.....	711
21.3.2. SQPI controller sampling polarity.....	712
21.3.3. SQPI controller special command.....	713
21.3.4. SQPI controller read ID command	713
21.3.5. SQPI controller output clock configuration.....	714
21.3.6. SQPI controller initialization	714
21.3.7. Read ID command flow.....	714
21.3.8. Read/Write operation flow	714
21.3.9. SQPI controller mode timing	714
21.4. Register definition	716
21.4.1. SQPI Initial Register (SQPI_INIT)	717
21.4.2. SQPI Read Command Register (SQPI_RCMD)	717
21.4.3. SQPI Write Command Register (SQPI_WCMD).....	718
21.4.4. SQPI ID Low Register (SQPI_IDL).....	719
21.4.5. SQPI ID High Register (SQPI_IDH).....	719
22. Quad-SPI interface (QSPI)	721
22.1. Overview	721
22.2. Characteristics.....	721
22.3. Function overview	721
22.3.1. QSPI block diagram.....	721
22.3.2. QSPI command format.....	723
22.3.3. QSPI signal line modes.....	724
22.3.4. CSN and SCK behavior.....	725

22.4. Operating modes	726
22.4.1. Indirect mode	726
22.4.2. Status polling mode.....	727
22.4.3. Memory map mode	727
22.4.4. FMC mode.....	728
22.5. QSPI configuration	729
22.5.1. Flash configuration.....	729
22.5.2. QSPI IP configuration	729
22.6. Security description.....	729
22.7. Send instruction only once.....	730
22.8. Busy	730
22.9. Error management	730
22.10. QSPI interrupts	731
22.11. Register definition	732
22.11.1. Control register (QSPI_CTL)	732
22.11.2. Device configuration register (QSPI_DCFG).....	734
22.11.3. Status register (QSPI_STAT).....	735
22.11.4. Status clear register (QSPI_STATC).....	737
22.11.5. Data length register (QSPI_DTLEN).....	738
22.11.6. Transfer configuration register (QSPI_TCFG)	738
22.11.7. Address register (QSPI_ADDR).....	740
22.11.8. Alternate bytes register (QSPI_ALTE).....	741
22.11.9. Data register (QSPI_DATA)	741
22.11.10. Secure Status register (QSPI_STAT_SEC).....	742
22.11.11. Secure Status clear register (QSPI_STATC_SEC).....	743
22.11.12. Secure Data length register (QSPI_DTLEN_SEC).....	744
22.11.13. Secure Transfer configuration register (QSPI_TCFG_SEC)	745
22.11.14. Secure Address register (QSPI_ADDR_SEC).....	747
22.11.15. Secure Alternate bytes register (QSPI_ALTE_SEC).....	747
22.11.16. Secure Data register (QSPI_DATA_SEC).....	748
22.11.17. Status mask register (QSPI_STATMK).....	748
22.11.18. Status match register (QSPI_STATMATCH)	749
22.11.19. Interval register (QSPI_INTERVAL).....	749
22.11.20. Timeout register (QSPI_TMOOUT).....	750
22.11.21. FIFO flush register (QSPI_FLUSH).....	750
22.11.22. Wait cnt for indirect wire mode register (QSPI_WTCNT).....	751
22.11.23. Timeout for status polling mode register (QSPI_SPTMOUT).....	751
22.11.24. FMC mode security configuration register (QSPI_FMC_SECCFG).....	751
22.11.25. Control register in FMC mode (QSPI_CTLF)	752
22.11.26. Transfer configuration register in FMC mode (QSPI_TCFGF).....	753
22.11.27. Alternate bytes register in FMC mode (QSPI_ALTEF).....	755

22.11.28. Complete bytes counter register (QSPI_BYTE_CNT).....	756
22.11.29. Privilege configuration register (QSPI_PRIVCFG)	756
23. Secure digital input/output interface (SDIO)	758
23.1. Introduction.....	758
23.2. Main features	758
23.3. SDIO bus topology.....	758
23.4. SDIO functional description	761
23.4.1. SDIO adapter	761
23.4.2. APB2 interface.....	765
23.5. Card functional description.....	766
23.5.1. Card registers.....	766
23.5.2. Commands	768
23.5.3. Responses.....	779
23.5.4. Data packets format	783
23.5.5. Two status fields of the card.....	784
23.6. Programming sequence	792
23.6.1. Card identification	792
23.6.2. No data commands	794
23.6.3. Single block or multiple block write.....	794
23.6.4. Single block or multiple block read.....	795
23.6.5. Stream write and stream read (MMC only).....	796
23.6.6. Erase.....	798
23.6.7. Bus width selection.....	799
23.6.8. Protection management	799
23.6.9. Card Lock/Unlock operation.....	799
23.7. Specific operations.....	802
23.7.1. SD I/O specific operations	802
23.7.2. CE-ATA specific operations.....	805
23.8. SDIO registers.....	807
23.8.1. Power control register (SDIO_PWRCTL)	807
23.8.2. Clock control register (SDIO_CLKCTL).....	807
23.8.3. Command argument register (SDIO_CMDAGMT).....	809
23.8.4. Command control register (SDIO_CMDCTL)	809
23.8.5. Command index response register (SDIO_RSPCMDIDX).....	811
23.8.6. Response register (SDIO_RESPx x=0..3).....	811
23.8.7. Data timeout register (SDIO_DATATO).....	812
23.8.8. Data length register (SDIO_DATALEN).....	812
23.8.9. Data control register (SDIO_DATACTL)	813
23.8.10. Data counter register (SDIO_DATACNT).....	814
23.8.11. Status register (SDIO_STAT)	815

23.8.12. Interrupt clear register (SDIO_INTC)	816
23.8.13. Interrupt enable register (SDIO_INTEN)	818
23.8.14. FIFO counter register (SDIO_FIFOCNT).....	819
23.8.15. FIFO data register (SDIO_FIFO).....	820
24. Universal serial bus full-speed interface (USBFS)	821
24.1. Overview	821
24.2. Characteristics.....	821
24.3. Block diagram.....	822
24.4. Signal description.....	822
24.5. Function overview	822
24.5.1. USBFS clocks and working modes	822
24.5.2. USB host function	824
24.5.3. USB device function.....	826
24.5.4. OTG function overview.....	827
24.5.5. Data FIFO.....	828
24.5.6. Operation guide.....	831
24.6. Interrupts.....	835
24.7. Register definition	837
24.7.1. Global control and status registers.....	837
24.7.2. Host control and status registers.....	858
24.7.3. Device control and status registers	870
24.7.4. Power and clock control register (USBFS_PWRCLKCTL).....	894
25. Digital camera interface (DCI)	895
25.1. Overview	895
25.2. Characteristics.....	895
25.3. Block diagram.....	895
25.4. Signal description.....	896
25.5. Function overview	896
25.5.1. DCI hardware synchronization mode.....	896
25.5.2. Embedded synchronization mode.....	897
25.5.3. Capture data using snapshot or continuous capture modes	897
25.5.4. Window function	898
25.5.5. Pixel formats, data padding and DMA.....	898
25.6. Interrupts.....	899
25.7. Register definition	900
25.7.1. Control register (DCI_CTL).....	900
25.7.2. Status register0 (DCI_STAT0)	901
25.7.3. Status register1 (DCI_STAT1)	902

25.7.4. Interrupt enable register (DCI_INTEN).....	902
25.7.5. Interrupt flag register (DCI_INTF).....	903
25.7.6. Interrupt flag clear register (DCI_INTC).....	904
25.7.7. Synchronization codes register (DCI_SC)	904
25.7.8. Synchronization codes unmask register (DCI_SCUMSK).....	905
25.7.9. Cropping window start position register (DCI_CWSPOS).....	905
25.7.10. Cropping window size register (DCI_CWSZ)	906
25.7.11. DATA register (DCI_DATA)	907
26. Touch sensing interface (TSI)	908
26.1. Overview	908
26.2. Characteristics.....	908
26.3. Function Overview.....	908
26.3.1. TSI block diagram	908
26.3.2. Touch sensing technique overview.....	908
26.3.3. Charge transfer sequence	909
26.3.4. Charge transfer sequence FSM.....	911
26.3.5. Clock and duration time of states.....	913
26.3.6. PIN mode control of TSI.....	914
26.3.7. Analog switch (ASW) and I/O hysteresis mode	914
26.3.8. TSI operation flow.....	915
26.3.9. TSI flags and interrupts	915
26.3.10. TSI GPIOs	915
26.4. Registers definition.....	916
26.4.1. Control register0 (TSI_CTL0).....	916
26.4.2. Interrupt enable register(TSI_INTEN).....	918
26.4.3. Interrupt flag clear register (TSI_INTC)	919
26.4.4. Interrupt flag register (TSI_INTF)	919
26.4.5. Pin hysteresis mode register(TSI_PHM).....	920
26.4.6. Analog switch register(TSI_ASW).....	920
26.4.7. Sample configuration register(TSI_SAMPCFG).....	921
26.4.8. Channel configuration register(TSI_CHCFG).....	921
26.4.9. Group control register(TSI_GCTL)	922
26.4.10. Group x cycle number registers(TSI_GxCYCN)(x = 0..2).....	923
26.4.11. Control register1 (TSI_CTL1)	923
27. Cryptographic Acceleration Unit (CAU)	925
27.1. Overview	925
27.2. Characteristics.....	925
27.3. CAU data type and initialization vectors	926
27.3.1. Data type.....	926
27.3.2. Initialization vectors	927

27.4. Cryptographic acceleration processor.....	927
27.4.1. DES/TDES cryptographic acceleration processor.....	928
27.4.2. AES cryptographic acceleration processor.....	932
27.5. Operating modes	939
27.6. CAU DMA interface	940
27.7. CAU interrupts.....	940
27.8. CAU suspended mode.....	941
27.9. Register definition	943
27.9.1. Control register (CAU_CTL).....	943
27.9.2. Status register 0 (CAU_STAT0).....	944
27.9.3. Data input register (CAU_DI)	945
27.9.4. Data output register (CAU_DO).....	946
27.9.5. DMA enable register (CAU_DMAEN)	946
27.9.6. Interrupt enable register (CAU_INTEN).....	947
27.9.7. Status register 1 (CAU_STAT1)	947
27.9.8. Interrupt flag register (CAU_INTF)	948
27.9.9. Key registers (CAU_KEY0..3(H/L))	949
27.9.10. Initial vector registers (CAU_IV0..1(H/L))	951
27.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x = 0..7).....	953
27.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x = 0..7)	953
28. Hash Acceleration Unit (HAU)	954
28.1. Overview	954
28.2. Characteristics.....	954
28.3. HAU data type	954
28.4. HAU core.....	956
28.4.1. Automatic data padding.....	956
28.4.2. Digest computing.....	957
28.4.3. Hash mode	958
28.4.4. HMAC mode	958
28.5. HAU suspended mode	959
28.5.1. Transfer data by CPU	959
28.5.2. Transfer data by DMA.....	959
28.6. HAU interrupt.....	960
28.6.1. Input FIFO interrupt.....	960
28.6.2. Calculation completion interrupt.....	960
28.7. Register definition	961
28.7.1. HAU control register (HAU_CTL).....	961
28.7.2. HAU data input register (HAU_DI)	962
28.7.3. HAU configuration register (HAU_CFG).....	963

28.7.4.	HAU data output register (HAU_DO0..7).....	964
28.7.5.	HAU interrupt enable register (HAU_INTEN).....	966
28.7.6.	HAU status and flag register (HAU_STAT)	966
28.7.7.	Context switch register x (HAU_CTXSx) (x = 0..53)	967
29.	Public Key Cryptographic Acceleration Unit (PKCAU)	968
29.1.	Overview	968
29.2.	Characteristics.....	968
29.3.	Function overview	968
29.3.1.	Operands.....	969
29.3.2.	RSA algorithm	969
29.3.3.	ECC algorithm	971
29.3.4.	Integer arithmetic operations.....	972
29.3.5.	Elliptic curve operations in Fp domain	982
29.3.6.	PKCAU operation process.....	988
29.3.7.	Processing times.....	989
29.3.8.	Status, errors and interrupts.....	990
29.4.	Register definition	992
29.4.1.	Control register (PKCAU_CTL).....	992
29.4.2.	Status register (PKCAU_STAT)	993
29.4.3.	Status clear register (PKCAU_STATC)	994
30.	High-Performance Digital Filter (HPDF)	996
30.1.	Overview	996
30.2.	Characteristics.....	996
30.3.	Function overview	996
30.3.1.	HPDF Block Diagram.....	996
30.3.2.	HPDF on-off control.....	997
30.3.3.	HPDF clock.....	998
30.3.4.	Multiplex serial data channel.....	999
30.3.5.	Parallel data input	1005
30.3.6.	Regular group conversion.....	1006
30.3.7.	Inserted group conversion	1007
30.3.8.	Digital filter.....	1009
30.3.9.	Integrator.....	1010
30.3.10.	Threshold monitor	1010
30.3.11.	Malfunction monitor.....	1013
30.3.12.	Extremes monitor.....	1013
30.3.13.	Data unit.....	1013
30.3.14.	HPDF interrupt.....	1015
30.4.	Register definition	1017
30.4.1.	HPDF channel x registers (x=0, 1)	1017

30.4.2. HPDF filter y registers (y=0, 1)	1022
31. Infrared ray port (IFRP)	1038
31.1. Overview	1038
31.2. Characteristics.....	1038
31.3. Function overview	1038
32. Wi-Fi	1040
32.1. Overview	1040
32.2. Characteristics.....	1040
32.2.1. Standards Supported.....	1040
32.2.2. Wi-Fi MAC	1040
32.2.3. Wi-Fi PHY.....	1040
32.2.4. Wi-Fi Radio.....	1041
33. Document appendix.....	1042
33.1. List of abbreviations used in registers.....	1042
33.2. List of terms.....	1042
33.3. Available peripherals	1042
34. Revision history	1043

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor	38
Figure 1-2. GD32W51x system architecture	40
Figure 1-3. Example of memory map security attribution vs SAU configuration regions ...	41
Figure 2-1. FMC bus in GD32W51x	67
Figure 2-2. Process of page erase operation.....	72
Figure 2-3. Process of mass erase operation.....	73
Figure 2-4. Process of word program operation.....	75
Figure 3-1. Block diagram of efuse controller	103
Figure 4-1. ICACHE block diagram	120
Figure 4-2. ICACHE remapping address	122
Figure 4-3. Power supply overview	132
Figure 4-4. Waveform of the POR / PDR.....	134
Figure 4-5. Waveform of the LVD threshold	134
Figure 4-6. RF sequence	139
Figure 6-1. The system reset circuit.....	153
Figure 6-2. Clock tree.....	154
Figure 6-3. HXTAL clock source	156
Figure 7-1. Block diagram of EXTI	220
Figure 8-1. Basic structure of a standard I/O port bit.....	232
Figure 8-2. Input configuration.....	233
Figure 8-3. Output configuration	234
Figure 8-4. Analog configuration.....	235
Figure 8-5. Alternate function configuration.....	235
Figure 9-1. Block diagram of TZPCU	252
Figure 10-1. Block diagram of CRC calculation unit.....	306
Figure 11-1. TRNG block diagram	310
Figure 12-1. Block diagram of DMA.....	316
Figure 12-2. Data stream for three transfer modes	317
Figure 12-3. Handshake mechanism.....	320
Figure 12-4. Data packing/unpacking when PWIDTH = '00'	326
Figure 12-5. Data packing/unpacking when PWIDTH = '01'	326
Figure 12-6. Data packing/unpacking when PWIDTH = '10'	327
Figure 12-7. DMA operation of switch-buffer mode	328
Figure 12-8. System connection of DMA0 and DMA1	336
Figure 14-1. ADC module block diagram	360
Figure 14-2. Single conversion mode.....	361
Figure 14-3. Continuous conversion mode	362
Figure 14-4. Scan conversion mode, continuous disable	363
Figure 14-5. Scan conversion mode, continuous enable	364
Figure 14-6. Discontinuous conversion mode	364

Figure 14-7. Auto-insertion, CTN = 1	365
Figure 14-8. Triggered insertion	366
Figure 14-9. Data alignment of 12-bit resolution	367
Figure 14-10. 20-bit to 16-bit result truncation	371
Figure 14-11. A numerical example with 5-bit shifting and rounding.....	371
Figure 15-1. Free watchdog timer block diagram.....	388
Figure 15-2. Window watchdog timer block diagram.....	394
Figure 15-3. Window watchdog timing diagram	395
Figure 16-1. Block diagram of RTC.....	399
Figure 16-2. Backup registers secure protections configuration	416
Figure 17-1. Advanced timer block diagram.....	448
Figure 17-2. Normal mode, internal clock divided by 1	449
Figure 17-3. Counter timing diagram with prescaler division change from 1 to 2	450
Figure 17-4. Timing chart of up counting mode, PSC=0/1	451
Figure 17-5. Timing chart of up counting mode, change TIMERx_CAR ongoing	451
Figure 17-6. Timing chart of down counting mode, PSC=0/1	452
Figure 17-7. Timing chart of down counting mode, change TIMERx_CAR ongoing	453
Figure 17-8. Timing chart of center-aligned counting mode	454
Figure 17-9. Repetition counter timing chart of center-aligned counting mode	455
Figure 17-10. Repetition counter timing chart of up counting mode	455
Figure 17-11. Repetition counter timing chart of down counting mode	456
Figure 17-12. Input capture logic.....	457
Figure 17-13. Output compare logic (with complementary output, x=0,1,2).....	458
Figure 17-14. Output compare logic (CH3_O)	458
Figure 17-15. Output-compare in three modes	460
Figure 17-16. Timing chart of EAPWM	461
Figure 17-17. Timing chart of CAPWM	461
Figure 17-18. Complementary output with dead-time insertion.....	464
Figure 17-19. Output behavior of the channel in response to a break (the break high active)	465
Figure 17-20. Example of counter operation in encoder interface mode	466
Figure 17-21. Example of encoder interface mode with CI0FE0 polarity inverted.....	466
Figure 17-22. Hall sensor is used to BLDC motor	467
Figure 17-23. Hall sensor timing between two timers	468
Figure 17-24. Restart mode	469
Figure 17-25. Pause mode	469
Figure 17-26. Event mode	470
Figure 17-27. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	471
Figure 17-28. Timer0 master/slave mode example	471
Figure 17-29. Trigger mode of TIMER0 controlled by enable signal of TIMER2	472
Figure 17-30. Trigger mode of TIMER0 controlled by update signal of TIMER2.....	473
Figure 17-31. Pause mode of TIMER0 controlled by enable signal of TIMER2	474
Figure 17-32. Pause mode of TIMER0 controlled by O0CPREF signal of TIMER2	474
Figure 17-33. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2.....	475

Figure 17-34. General Level 0 timer block diagram.....	506
Figure 17-35. Normal mode, internal clock divided by 1.....	507
Figure 17-36. Counter timing diagram with prescaler division change from 1 to 2.....	508
Figure 17-37. Timing chart of up counting mode, PSC=0/1	509
Figure 17-38. Timing chart of up counting mode, change TIMERx_CAR ongoing	509
Figure 17-39. Timing chart of down counting mode,PSC=0/1	510
Figure 17-40. Timing chart of down counting mode, change TIMERx_CAR ongoing	511
Figure 17-41. Timing chart of center-aligned counting mode.....	512
Figure 17-42. Input capture logic.....	513
Figure 17-43. Output compare logic (x=0,1,2,3).....	514
Figure 17-44. Output-compare in three modes	515
Figure 17-45. Timing chart of EAPWM	516
Figure 17-46. Timing chart of CAPWM.....	516
Figure 17-47. Example of counter operation in encoder interface mode	518
Figure 17-48. Example of encoder interface mode with CI0FE0 polarity inverted.....	518
Figure 17-49. Restart mode	519
Figure 17-50. Pause mode	520
Figure 17-51. Event mode	520
Figure 17-52. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	521
Figure 17-53. General level4 timer block diagram	550
Figure 17-54. Normal mode, internal clock divided by 1.....	551
Figure 17-55. Counter timing diagram with prescaler division change from 1 to 2.....	551
Figure 17-56. Up-counter timechart, PSC=0/1	552
Figure 17-57. Up-counter timechart, change TIMERx_CAR on the go	553
Figure 17-58. Repetition timechart for up-counter	554
Figure 17-59. Input capture logic.....	555
Figure 17-60. Output compare logic (with complementary output, x=0)	556
Figure 17-61. Output-compare under three modes	557
Figure 17-62. PWM mode timechart	558
Figure 17-63. Complementary output with dead-time insertion.....	560
Figure 17-64. Output behavior in response to a break(The break high active).....	561
Figure 17-65. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	562
Figure 17-66. Basic timer block diagram	580
Figure 17-67. Normal mode, internal clock divided by 1.....	581
Figure 17-68. Counter timing diagram with prescaler division change from 1 to 2.....	581
Figure 17-69. Timing chart of up counting mode, PSC=0/1	582
Figure 17-70. Timing chart of up counting mode, change TIMERx_CAR ongoing	583
Figure 18-1. USART module block diagram.....	591
Figure 18-2. USART character frame (8 bits data and 1 stop bit)	592
Figure 18-3. USART transmit procedure	594
Figure 18-4. Oversampling method of a receive frame bit (OSB=0)	595
Figure 18-5. Configuration step when using DMA for USART transmission	596
Figure 18-6. Configuration step when using DMA for USART reception	597
Figure 18-7. Hardware flow control between two USARTs	597

Figure 18-8. Hardware flow control	598
Figure 18-9. Break frame occurs during idle state	599
Figure 18-10. Break frame occurs during a frame	600
Figure 18-11. Example of USART in synchronous mode	600
Figure 18-12. 8-bit format USART synchronous waveform (CLEN=1)	601
Figure 18-13. IrDA SIR ENDEC module	601
Figure 18-14. IrDA data modulation	602
Figure 18-15. ISO7816-3 frame format	603
Figure 18-16. USART Receive FIFO structure	605
Figure 18-17. USART interrupt mapping diagram	607
Figure 19-1. I2C module block diagram	629
Figure 19-2. Data validation	630
Figure 19-3. START and STOP condition	631
Figure 19-4. I2C communication flow with 10-bit address (Master Transmit)	631
Figure 19-5. I2C communication flow with 7-bit address (Master Transmit)	632
Figure 19-6. I2C communication flow with 7-bit address (Master Receive)	632
Figure 19-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)	632
Figure 19-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)	632
Figure 19-9. Data hold time	633
Figure 19-10. Data setup time	634
Figure 19-11. Data transmission	636
Figure 19-12. Data reception	636
Figure 19-13. I2C initialization in slave mode	640
Figure 19-14. Programming model for slave transmitting when SS=0	641
Figure 19-15. Programming model for slave transmitting when SS=1	642
Figure 19-16. Programming model for slave receiving	643
Figure 19-17. I2C initialization in master mode	644
Figure 19-18. Programming model for master transmitting ($N \leq 255$)	645
Figure 19-19. Programming model for master transmitting ($N > 255$)	646
Figure 19-20. Programming model for master receiving ($N \leq 255$)	647
Figure 19-21. Programming model for master receiving ($N > 255$)	648
Figure 19-22. SMBus Master Transmitter and Slave Receiver communication flow	652
Figure 19-23. SMBus Master Receiver and Slave Transmitter communication flow	653
Figure 20-1. Block diagram of SPI	671
Figure 20-2. SPI timing diagram in normal mode	673
Figure 20-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)	673
Figure 20-4. A typical full-duplex connection	675
Figure 20-5. A typical simplex connection (Master: Receive, Slave: Transmit)	675
Figure 20-6. A typical simplex connection (Master: Transmit only, Slave: Receive)	676
Figure 20-7. A typical bidirectional connection	676
Figure 20-8. Timing diagram of TI master mode with discontinuous transfer	678
Figure 20-9. Timing diagram of TI master mode with continuous transfer	678

Figure 20-10. Timing diagram of TI slave mode.....	679
Figure 20-11. Timing diagram of quad write operation in Quad-SPI mode.....	680
Figure 20-12. Timing diagram of quad read operation in Quad-SPI mode.....	681
Figure 20-13. Block diagram of I2S.....	684
Figure 20-14. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)	686
Figure 20-15. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)	686
Figure 20-16. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)	686
Figure 20-17. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)	686
Figure 20-18. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)	686
Figure 20-19. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)	687
Figure 20-20. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)	687
Figure 20-21. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)	687
Figure 20-22. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)	687
Figure 20-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)	688
Figure 20-24. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)	688
Figure 20-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)	688
Figure 20-26. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)	688
Figure 20-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)	688
Figure 20-28. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)	688
Figure 20-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)	689
Figure 20-30. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)	689
Figure 20-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)	689
Figure 20-32. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)	689
Figure 20-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)	690
Figure 20-34. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)	690
Figure 20-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)	690
Figure 20-36. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)	690
Figure 20-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)	690
Figure 20-38. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)	691
Figure 20-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)	691
Figure 20-40. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)	691
Figure 20-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)	691
Figure 20-42. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)	691
Figure 20-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)	692

Figure 20-44. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)	692
Figure 20-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)	692
Figure 20-46. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)	692
Figure 20-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)	692
Figure 20-48. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)	693
Figure 20-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)	693
Figure 20-50. Block diagram of I2S clock generator	693
Figure 21-1. SQPI Polarity Example	712
Figure 21-2. SQPI SCMD Example	713
Figure 21-3. SQPI Read ID Example (IDLEN=00)	713
Figure 21-4. SQPI_CLK Example	714
Figure 21-5. SQPI SSS Mode Timing (SPI).....	715
Figure 21-6. SQPI SSQ Mode Timing	715
Figure 21-7. SQPI SQQ Mode Timing (SQPI).....	715
Figure 21-8. SQPI QQQ Mode Timing (QPI).....	716
Figure 21-9. SQPI SSD Mode Timing	716
Figure 21-10. SQPI SDD Mode Timing.....	716
Figure 22-1. QSPI diagram.....	722
Figure 22-2. QSPI command format.....	723
Figure 22-3. CSN and SCK behavior.....	726
Figure 23-1. SDIO “no response” and “no data” operations	759
Figure 23-2. SDIO multiple blocks read operation.....	760
Figure 23-3. SDIO multiple blocks write operation.....	760
Figure 23-4. SDIO sequential read operation.....	760
Figure 23-5. SDIO sequential write operation.....	760
Figure 23-6. SDIO block diagram.....	761
Figure 23-7. Command Token Format.....	768
Figure 23-8. Response Token Format.....	780
Figure 23-9. 1-bit data bus width	784
Figure 23-10. 4-bit data bus width	784
Figure 23-11. 8-bit data bus width	784
Figure 23-12. Read wait control by stopping SDIO_CK	802
Figure 23-13. Read wait operation using SDIO_D[2]	803
Figure 23-14. Function2 read cycle inserted during function1 multiple read cycle	803
Figure 23-15. Read Interrupt cycle timing	804
Figure 23-16. Write interrupt cycle timing	804
Figure 23-17. Multiple block 4-Bit read interrupt cycle timing	805
Figure 23-18. Multiple block 4-Bit write interrupt cycle timing.....	805

Figure 23-19. The operation for command completion disable signal	806
Figure 24-1. USBFS block diagram.....	822
Figure 24-2. Connection with host or device mode	823
Figure 24-3. Connection with OTG mode.....	824
Figure 24-4. State transition diagram of host port.....	824
Figure 24-5. HOST mode FIFO space in SRAM	829
Figure 24-6. Host mode FIFO access register map.....	829
Figure 24-7. Device mode FIFO space in SRAM	830
Figure 24-8. Device mode FIFO access register map.....	831
Figure 25-1. DCI module block diagram.....	895
Figure 25-2. Hardware synchronization mode.....	896
Figure 25-3. Hardware synchronization mode: JPEG format supporting	897
Figure 26-1. Block diagram of TSI module	908
Figure 26-2. Block diagram of Sample pin and Channel Pin.....	909
Figure 26-3. Voltage of a sample pin during charge-transfer sequence	911
Figure 26-4. FSM flow of a charge-transfer sequence	912
Figure 27-1. DATAM No swapping and Half-word swapping	926
Figure 27-2. DATAM Byte swapping and Bit swapping.....	927
Figure 27-3. CAU diagram.....	928
Figure 27-4. DES/TDES ECB encryption	929
Figure 27-5. DES/TDES ECB decryption	930
Figure 27-6. DES/TDES CBC encryption	931
Figure 27-7. DES/TDES CBC decryption	932
Figure 27-8. AES ECB encryption	933
Figure 27-9. AES ECB decryption	933
Figure 27-10. AES CBC encryption.....	934
Figure 27-11. AES CBC decryption.....	935
Figure 27-12. Counter block structure	935
Figure 27-13. AES CTR encryption/decryption	936
Figure 28-1. DATAM No swapping and Half-word swapping	955
Figure 28-2. DATAM Byte swapping and Bit swapping.....	955
Figure 28-3. HAU block diagram.....	956
Figure 29-1. PKCAU module block diagram.....	969
Figure 29-2. Flow chart of RSA algorithm.....	970
Figure 29-3. Flow chart of ECDSA sign	971
Figure 29-4. Flow chart of ECDSA verification	972
Figure 29-5. Arithmetic addition.....	973
Figure 29-6. Arithmetic subtraction	974
Figure 29-7. Arithmetic multiplication.....	974
Figure 29-8. Arithmetic comparison	975
Figure 29-9. Modular reduction	975
Figure 29-10. Modular addition.....	976
Figure 29-11. Modular subtraction	977
Figure 29-12. Montgomery parameter calculation.....	977

Figure 29-13. Mutual mapping between Montgomery domain and natural domain.....	978
Figure 29-14. Montgomery multiplication.....	979
Figure 29-15. Modular exponentiation of normal mode	979
Figure 29-16. Modular exponentiation of fast mode	980
Figure 29-17. Modular inversion.....	980
Figure 29-18. RSA CRT exponentiation	981
Figure 29-19. Point on elliptic curve Fp check	983
Figure 29-20. ECC scalar multiplication of normal mode	984
Figure 29-21. ECC scalar multiplication of fast mode	984
Figure 29-22. ECDSA sign	986
Figure 29-23. ECDSA verification.....	987
Figure 30-1. HPDF block diagram	997
Figure 30-2. The sequence diagram of SPI data transmission	1000
Figure 30-3. The sequence diagram of Manchester data transmission	1001
Figure 30-4. Manchester synchronous sequence diagram.....	1002
Figure 30-5. Clock loss detection timing diagram.....	1003
Figure 30-6. Channel pins redirection.....	1004
Figure 30-7. HPDF module external input data processing flow	1014
Figure 30-8. HPDF interrupt logic diagram	1016
Figure 31-1. IFRP output timechart 1.....	1038
Figure 31-2. IFRP output timechart 2.....	1039
Figure 31-3. IFRP output timechart 3.....	1039

List of Tables

Table 1-1. The interconnection relationship of the AHB interconnect matrix	39
Table 1-2. Default system security state	41
Table 1-3. Securable peripherals by TZSPC	42
Table 1-4. TrustZone-aware peripherals.....	43
Table 1-5. Memory map based on IDAU mapping of GD32W51x devices	44
Table 1-6. BOOT0 modes.....	48
Table 1-7. BOOT1 modes.....	48
Table 1-8. Boot address modes when TrustZone is disabled (TZEN=0)	49
Table 1-9. Boot modes when TrustZone is enabled (TZEN=1)	49
Table 2-1. GD32W51x base address and size for flash memory (FMC mode).....	68
Table 2-2. GD32W51x base address and size for flash memory (QSPI mode).....	69
Table 2-3. Option bytes.....	75
Table 2-4. Flash secure operation under different protection levels when TrustZone is active (TZEN=1).....	79
Table 2-5. Flash non-secure operation under different protection levels when TrustZone is active (TZEN=1)	80
Table 2-6. Flash operation under different protection levels when TrustZone is disable (TZEN=0).....	80
Table 2-7. Flash mass erase operation under different protection levels when TrustZone is active (TZEN=1)	81
Table 2-8. Flash interrupt requests (non-secure).....	82
Table 2-9. Flash interrupt requests (secure)	83
Table 3-1. EFUSE address mapping.....	103
Table 3-2. system parameters.....	104
Table 4-1. TAG memory parameters.....	121
Table 4-2. ICAHCE remap region size	122
Table 4-3. ICACHE cache ability for transaction.....	123
Table 4-4. memory configuration.....	123
Table 4-5. ICACHE interrupt.....	125
Table 4-6 Typical work mode	138
Table 4-7. Time in RF sequence	139
Table 4-8. Power saving mode summary.....	140
Table 4-9. PMU Security configuration summary.....	141
Table 6-1. Clock output 0 source select	159
Table 6-2. Clock output 1 source select	159
Table 6-3. 1.2V domain voltage selected in deep-sleep mode	159
Table 6-4. RCU secure protection configuration summary.....	160
Table 6-5. RCU secure-bit or nonsecure-bit access rules	161
Table 6-6. RCU register privileg and unprivileg access rules.....	161
Table 7-1. NVIC exception types in Cortex-M33.....	217

Table 7-2. Interrupt vector table	217
Table 7-3. EXTI source	221
Table 7-4. Register protection overview	223
Table 8-1. GPIO configuration table	231
Table 8-2. GPIO secure state	236
Table 8-3. GPIOx_CTL reset value	237
Table 8-4. GPIOx_OSPD reset value	241
Table 8-5. GPIOx_PUD reset value	243
Table 8-6. GPIOx_AFSEL0 reset value	247
Table 8-7. GPIOx_AFSEL1 reset value	248
Table 9-1. TrustZone® peripherals	253
Table 9-2. TZMMPCx	255
Table 9-3. TZBMPcx	256
Table 9-4. Trace and debug state	257
Table 9-5. SPC/GSSA debug	257
Table 12-1. Transfer mode	317
Table 12-2. Peripheral requests to DMA0	320
Table 12-3. Peripheral requests to DMA1	321
Table 12-4. CNT configuration	323
Table 12-5. FIFO counter critical value configuration rules	324
Table 12-6. DMA interrupt events	333
Table 14-1. ADC internal signals	359
Table 14-2. ADC pins definition	359
Table 14-3. External trigger modes	367
Table 14-4. External trigger for regular channels of ADC	368
Table 14-5. External trigger for inserted channels of ADC	368
Table 14-6. Maximum output results for N and M combinations (grayed values indicates truncation)	371
Table 15-1. Min/max FWDGT timeout period at 32 kHz (IRC32K)	388
Table 15-2. Min-max timeout value at 45 MHz (f_{PCLK1})	395
Table 16-1. RTC register secure access rules	411
Table 16-2. RTC secure mode configuration summary	411
Table 16-3. RTC register privilege access rules	413
Table 16-4. RTC privileged protected mode configuration summary	414
Table 16-5. RTC power saving mode management	417
Table 16-6. RTC non-secure interrupts control	417
Table 16-7. RTC secure interrupts control	418
Table 17-1. Timers (TIMERx) are divided into four sorts	446
Table 17-2. Complementary outputs controlled by parameters	463
Table 17-3. Counting direction versus encoder signals	466
Table 17-4. Examples of slave mode	468
Table 17-5. Counting direction versus encoder signals	517
Table 17-6. Examples of slave mode	519
Table 17-7. Complementary outputs controlled by parameters	559

Table 18-1. Description of USART important pins	591
Table 18-2. Configuration of stop bits	592
Table 18-3. USART interrupt requests	606
Table 19-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)	629
Table 19-2. Data setup time and data hold time	635
Table 19-3. Communication modes to be shut down	637
Table 19-4. I2C configuration	637
Table 19-5. SMBus with PEC configuration	650
Table 19-6. I2C error flags	654
Table 19-7. I2C interrupt events	654
Table 20-1. SPI signal description	671
Table 20-2. Quad-SPI signal description	672
Table 20-3. SPI operating modes	674
Table 20-4. SPI interrupt requests	684
Table 20-5. I2S bitrate calculation formulas	693
Table 20-6. Audio sampling frequency calculation formulas	694
Table 20-7. Direction of I2S interface signals for each operation mode	694
Table 20-8. I2S interrupt	699
Table 21-1. SQPI controller mode definition	711
Table 22-1. QSPI signal description	722
Table 22-2. QSPI singnal line modes	724
Table 22-3. Flash secure/non-secure and privileged/unprivileged operation under FMC mode when TrustZone is active (TZEN=1)	729
Table 22-4. QSPI interrupt requests	731
Table 23-1. SDIO I/O definitions	762
Table 23-2. Command format	768
Table 23-3. Card command classes (CCCs)	769
Table 23-4. Basic commands (class 0)	771
Table 23-5. Block-Oriented read commands (class 2)	773
Table 23-6. Stream read commands (class 1) and stream write commands (class 3)	774
Table 23-7. Block-Oriented write commands (class 4)	774
Table 23-8. Erase commands (class 5)	775
Table 23-9. Block oriented write protection commands (class 6)	776
Table 23-10. Lock card (class 7)	776
Table 23-11. Application-specific commands (class 8)	777
Table 23-12. I/O mode commands (class 9)	778
Table 23-13. Switch function commands (class 10)	779
Table 23-14. Response R1	781
Table 23-15. Response R2	781
Table 23-16. Response R3	781
Table 23-17. Response R4 for MMC	782
Table 23-18. Response R4 for SD I/O	782
Table 23-19. Response R5 for MMC	782

Table 23-20. Response R5 for SD I/O	782
Table 23-21. Response R6	783
Table 23-22. Response R7	783
Table 23-23. Card status	785
Table 23-24. SD status	788
Table 23-25. Performance move field	790
Table 23-26. AU_SIZE field	790
Table 23-27. Maximum AU size	791
Table 23-28. Erase size field	791
Table 23-29. Erase timeout field	791
Table 23-30. Erase offset field	792
Table 23-31. Lock card data structure	800
Table 23-32. SDIO_RESPx register at different response type	811
Table 24-1. USBFS signal description	822
Table 24-2. USBFS global interrupt	835
Table 25-1. PINs used by DCI	896
Table 25-2. Memory view in byte padding mode	898
Table 25-3. Memory view in half-word padding mode	899
Table 25-4. Status/Error flags	899
Table 26-1. Pin and analog switch state in a charge-transfer sequence	910
Table 26-2. Duration time of Extend Charge state in each cycle	913
Table 26-3. Spread spectrum deviation base on HCLK period	914
Table 26-4. TSI errors and flags	915
Table 26-5. TSI pins	915
Table 29-1. Parameters of RSA algorithm	970
Table 29-2. Integer arithmetic operations	972
Table 29-3. Range of parameters used by RSA CRT exponentiation operation	982
Table 29-4. Elliptic curve operations in Fp domain	982
Table 29-5. Range of parameters used by point on elliptic curve Fp check	983
Table 29-6. Range of parameters used by ECC scalar multiplication	985
Table 29-7. Range of parameters used by ECDSA sign	986
Table 29-8. Range of parameters used by ECDSA verification	988
Table 29-9. Modular exponentiation computation times	989
Table 29-10. ECC scalar multiplication computation times	990
Table 29-11. ECDSA signature average computation times	990
Table 29-12. ECDSA verification average computation times	990
Table 29-13. Montgomery parameters average computation times	990
Table 29-14. PKCAU interrupt requests	991
Table 30-1. HPDF pins definition	997
Table 30-2. SPI interface clock configuration	999
Table 30-3. Parallel data packed mode	1006
Table 30-4. Trigger signal of inserted group	1008
Table 30-5. The relationship between the maximum output resolution and oversampling filtering of Sinc^X filtering	1009

Table 30-6. Relationship between the maximum output resolution and IOR, SFOR, SFO of the integrator.....	1010
Table 30-7. Features of threshold monitor working mode.....	1011
Table 30-8. Maximum output rate.....	1014
Table 30-9. HPDF interrupt event.....	1015
Table 33-1. List of abbreviations used in register.....	1042
Table 33-2. List of terms.....	1042
Table 34-1. Revision history.....	1043

1. System and memory architecture

The devices of GD32W51x series are highly integrated 2.4GHz Wi-Fi System-on-Chip (SoC) 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M33 processor with Trustzone. 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.

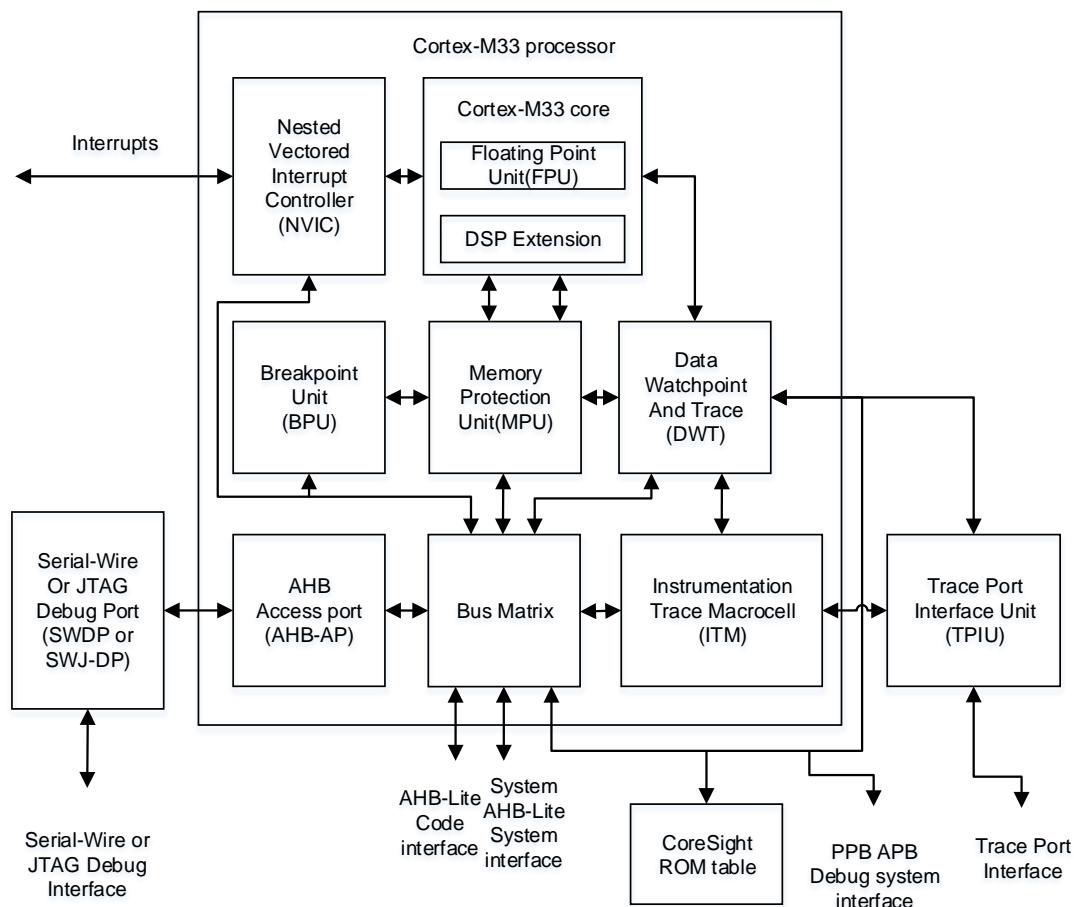
1.1. Arm Cortex-M33 processor

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)
- Arm® TrustZone® technology, using the ARMv8-M main extension supporting secure and non-secure states
- Memory Protection Unit (MPU), supporting 8 regions for secure and 8 regions for non-secure
- Configurable secure attribute unit (SAU) supporting up to 8 memory regions
- Floating Point Unit (FPU)
- DSP Extension (DSP)

[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 GD32W51x devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, three AHB buses 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 ‘0’ means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix.

Table 1-1. The interconnection relationship of the AHB interconnect matrix

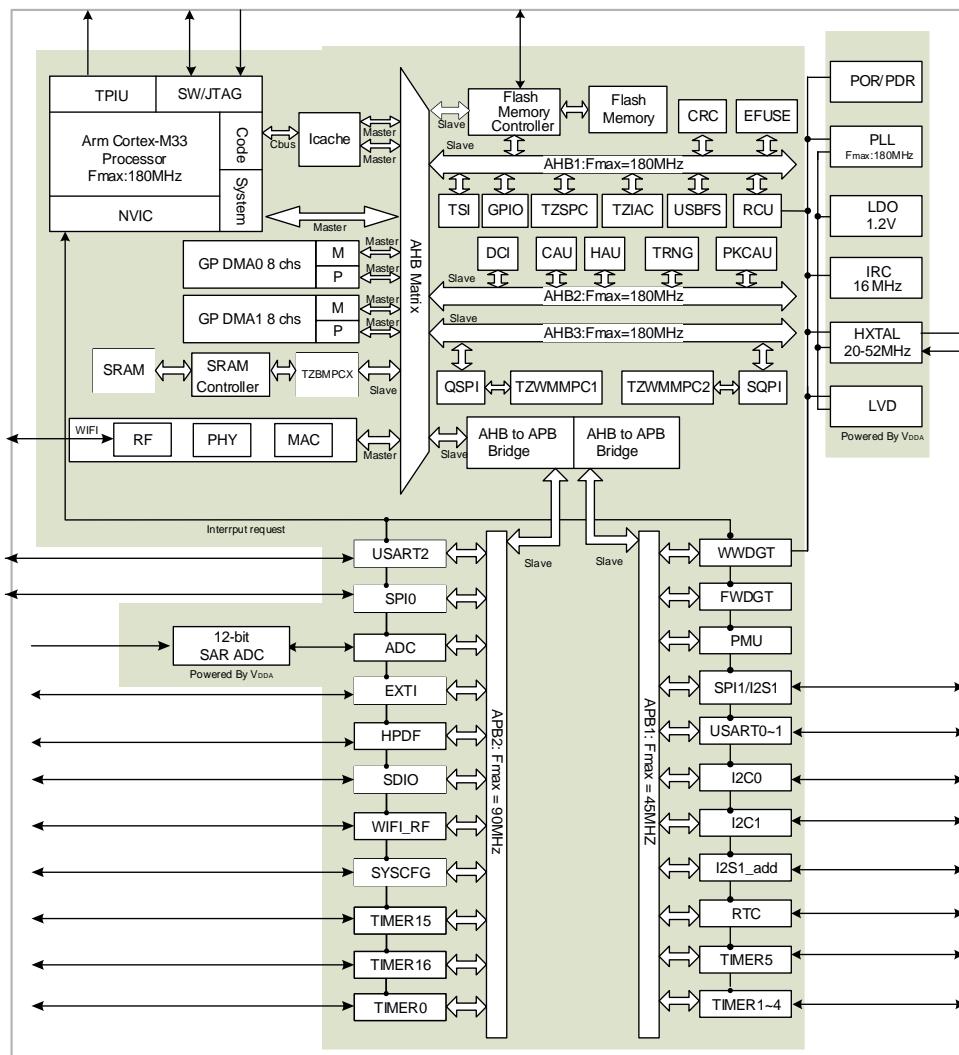
	S-CBUS	F-CBUS	SBUS	DMA0M	DMA1M	Wi-Fi	DMA0P	DMA1P
FMC	0	1	0	1	1	0	0	1
SRAM0	0	1	1	1	1	1	0	1
AHB1	0	0	1	0	1	0	1	1
AHB2	0	0	1	0	1	0	0	1
AHB3	1	0	1	1	1	1	0	1
SRAM1	0	1	1	1	1	1	0	1
SRAM2	0	1	1	1	1	1	0	1
SRAM3	0	1	1	1	1	1	0	1
APB1	0	0	1	0	1	0	1	1
APB2	0	0	1	0	1	0	1	1

As is shown above, there are several masters connected with the AHB interconnect matrix, including S-CBUS, F-CBUS, SBUS, DMA0M, DMA0P DMA1M, DMA1P and Wi-Fi. S-CBUS and F-CBUS is the code bus of the Cortex®-M33 core. F-CBUS is used for instruction fetch and data access to the internal memories mapped in code region, the target of F-CBUS are the internal Flash and internal SRAMs. S-CBUS is used for instruction fetch and data access to the external memories mapped in code region, the target of this bus are the external memories (QSPI_flash and SQPI_PSRAM). 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. DMA0M and DMA1M are the memory buses of DMA0 and DMA1 respectively, which is used by the DMA to perform transfer to/from memories, and the targets of DMA0M bus are internal Flash, internal SRAMs and external memories (QSPI_flash and SQPI_PSRAM), the targets of DMA1M bus are internal Flash, internal SRAMs, external memories (QSPI_flash and SQPI_PSRAM) and the AHB/APB peripherals. DMA0P and DMA1P are the peripheral buses of DMA0 and DMA1 respectively, which is used by the DMA to access AHB/APB peripherals or to perform memory-to-memory transfers. The targets of DMA0P bus are the AHB/APB peripherals, the targets of DMA1P bus are internal Flash, internal SRAMs, external memories (QSPI_flash and SQPI_PSRAM) and the AHB/APB peripherals. Wi-Fi is the bus connects the AHB master interface of the Wi-Fi to the BusMatrix, the targets of this bus are the internal Flash, internal SRAMs, Peripheral region and the external memories (QSPI_flash and SQPI_PSRAM).

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM0, SRAM1, SRAM2, SRAM3, AHB1, AHB2, AHB3, APB1 and APB2. FMC is the bus interface of the flash memory controller. SRAM0~SRAM3 is on-chip static random access memories. AHB1 is the AHB bus connected with all of the AHB1 slaves. AHB2 is the AHB bus connected with AHB2 slaves. AHB3 is the bus interface of the QSPI_flash and SQPI_PSRAM memory controller. 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 limited to 45Mhz, APB2 is limited to 90Mhz.

These are interconnected using a multilayer AHB bus architecture as shown in [Figure 1-2. GD32W51x system architecture](#).

Figure 1-2. GD32W51x system architecture



1.3. TrustZone® overview

1.3.1. TrustZone® security attribution

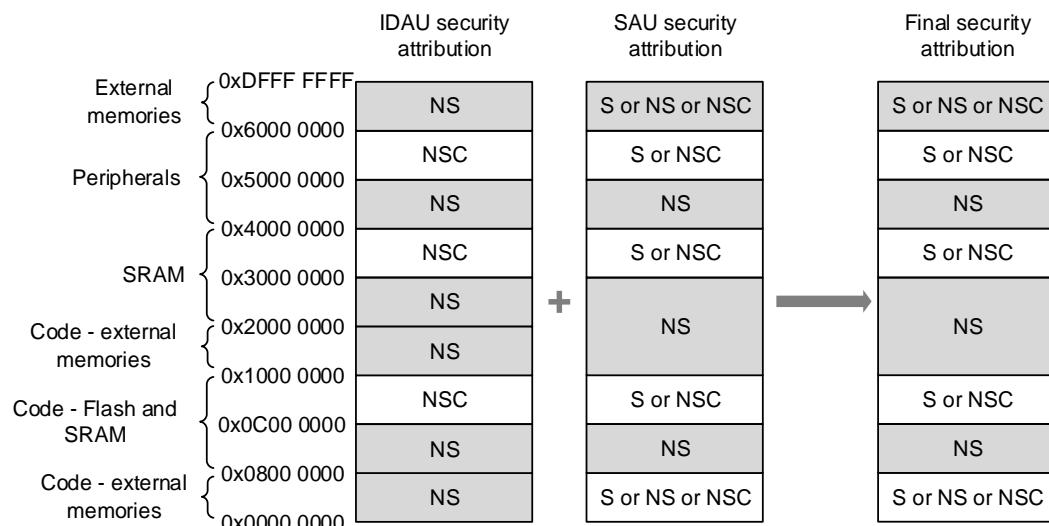
The security architecture is based on Arm® TrustZone® with the ARMv8-M Main Extension. The TrustZone security is activated by the TZEN option bit in the EFUSE_TZCTL register or the TZEN option bit in the option byte.

When the TrustZone is enabled, the SAU (security attribution unit) and IDAU (implementation defined attribution unit) are used to define security attribution of memory addresses. Up to eight SAU configurable regions are available for security attribution, the security attribution of each memory region can be set to secure(S), non-secure (NS), or non-secure callable (NSC) by SAU. IDAU predefines a fixed security attribution partition as non-secure (NS) and non-

secure callable (NSC), the IDAU memory map partition is not configurable and fixed by hardware implementation (refer to [Table 1-5. Memory map based on IDAU mapping of GD32W51x devices](#)). However the SAU can change security attribution of the memory by software. The security attribution of a memory address is determined by IDAU and SAU together. If the definitions are different, the attribution with a higher security level will be used (S > NSC > NS). [Figure 1-3. Example of memory map security attribution vs SAU configuration regions](#) shows an example of typical eight SAU regions mapping based on IDAU regions. The user can split and choose the secure, non-secure or NSC regions for external memories as needed.

Based on IDAU security attribution, the Flash, system SRAM0~3 and peripherals memory space is aliased twice for secure and non-secure state. However, the external memories (QSPI_FLASH and SQPI_PSRAM) space is not aliased.

Figure 1-3. Example of memory map security attribution vs SAU configuration regions



When the TrustZone security is activated by the TZEN option bit in the EFUSE_TZCTL register or the TZEN option bit in option byte, the default system security refer to [Table 1-2. Default system security state](#):

Table 1-2. Default system security state

	security state
CPU	Cortex-M33 is in secure state after reset. The boot address must be in secure address.
Memory map	SAU is fully secure after reset. Consequently, all memory map is fully secure. Up to 8 SAU configurable regions are available for security attribution.
Flash	Flash is secure after reset.

SRAMs	All SRAMs are secure after reset. TZBMPc (TrustZone block-based memory protection controller) is secure.
External memories	QSPI_FLASH and SQPI_PSRAM banks are secure after reset.
Peripherals	Securable peripherals are non-secure after reset. TrustZone-aware peripherals are non-secure after reset (except GPIOA, GPIOB and GPIOC). Their secure configuration registers are secure. Refer to Table 1-3. Securable peripherals by TZSPC and Table 1-4. TrustZone-aware peripherals for a list of Securable and TrustZone-aware peripherals.
GPIO	All GPIO are secure after reset
Interrupts	NVIC: All interrupts are secure after reset. NVIC is banked for secure and non-secure state. TZIAC: All illegal access interrupts are disabled after reset.

1.3.2. Peripheral classification

When the TrustZone security is active, a peripheral can be securable, secure, non-secure or TrustZone-aware type. For securable peripherals by TZSPC (TrustZone security controller), the SEC security bit corresponding to this peripheral is set in the TZSPC_SAM_CFGx register. For securable peripherals by TZSPC (TrustZone security controller), the SEC security bit corresponding to this peripheral is set in the TZSPC_SAM_CFGx register. TZIAC, TZBMPcx and EFUSE are always secure and AHB masters (Wi-Fi) always non-secure. For TrustZone-aware peripherals, a security feature of this peripheral is enabled through its dedicated bits.

[Table 1-3. Securable peripherals by TZSPC](#) and [Table 1-4. TrustZone-aware peripherals](#) summarize the list of securable and TrustZone-aware peripherals within the system.

Table 1-3. Securable peripherals by TZSPC

BUS	Peripheral
AHB1	CRC
	USBFS
	Wi-Fi
	ICACHE
	QSPI_FLASH(REG)
	SQPI_PSRAM(REG)
	TSI
AHB2	DCI
	CAU
	HAU
	TRNG
	PKCAU
AHB3	SQPI_PSRAM
APB1	TIMER1

	TIMER2
	TIMER3
	TIMER4
	TIMER5
	WWDGT
	FWDGT
	SPI1/I2S1
	USART0
	USART1
	I2C0
	I2C1
	I2S1_add
APB2	ADC
	TIMER0
	SPI0
	USART2
	TIMER15
	TIMER16
	Wi-Fi_RF
	SDIO
	HPDF

Table 1-4. TrustZone-aware peripherals

BUS	Peripheral
AHB1	DMA0
	DMA1
	RCU
	FMC
	GPIOA
	GPIOB
	GPIOC
	TZSPC
AHB3	QSPI_FLASH
APB1	RTC
	PMU
APB2	EXTI
	SYSCFG

1.4. 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 1-5. Memory map based on IDAU mapping of GD32W51x devices**](#) shows the memory map of the GD32W51x 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 1-5. Memory map based on IDAU mapping of GD32W51x devices

Pre-defined Regions	Bus	Secure boundary address	Non-Secure boundary address	Peripherals
	-	-	0xE000 1000 - 0xE00F FFFF	Cortex M33 internal peripherals
External device	AHB3	-	0x9800 0000 - 0xDFFF FFFF	Reserved
		-	0x9000 0000 - 0x97FF FFFF	QSPI_FLASH(MEM)
		-	0x6800 0000 - 0x8FFF FFFF	Reserved
		-	0x6000 0000 - 0x67FF FFFF	SQPI_PSRAM(MEM)
		0x5C06 3000 - 0x5FFF FFFF	0x4C06 3000 - 0x4FFF FFFF	Reserved
Peripheral	AHB2	0x5C06 1000 - 0x5C06 2FFF	0x4C06 1000 - 0x4C06 2FFF	PKCAU
		0x5C06 0C00 - 0x5C06 0FFF	0x4C06 0C00 - 0x4C06 0FFF	Reserved
		0x5C06 0800 - 0x5C06 0BFF	0x4C06 0800 - 0x4C06 0BFF	TRNG
		0x5C06 0400 - 0x5C06 07FF	0x4C06 0400 - 0x4C06 07FF	HAU
		0x5C06 0000 - 0x5C06 03FF	0x4C06 0000 - 0x4C06 03FF	CAU
		0x5C05 0400 - 0x5C05 FFFF	0x4C05 0400 - 0x4C05 FFFF	Reserved
		0x5C05 0000 - 0x5C05 03FF	0x4C05 0000 - 0x4C05 03FF	DCI
		0x5C04 0000 - 0x5C04 FFFF	0x4C04 0000 - 0x4C04 FFFF	Reserved
		0x5C00 0000 - 0x5C03 FFFF	0x4C00 0000 - 0x4C03 FFFF	Reserved
	AHB1	0x5904 0000 - 0x5BFF FFFF	0x4904 0000 - 0x4BFF FFFF	Reserved
		0x5900 0000 - 0x5903 FFFF	0x4900 0000 - 0x4903 FFFF	USBFS
		0x500B 1000 - 0x58FF FFFF	0x400B 1000 - 0x48FF FFFF	Reserved
		0x500B 0800 - 0x500B 0FFF	0x400B 0800 - 0x400B 0FFF	Reserved
		0x500B 0400 - 0x500B 07FF	0x400B 0400 - 0x400B 07FF	TZBMPC3
		0x500B 0000 - 0x500B 03FF	0x400B 0000 - 0x400B 03FF	TZBMPC2
		0x500A 1000 - 0x500A FFFF	0x400A 1000 - 0x400A FFFF	Reserved
		0x500A 0C00 - 0x500A 0FFF	0x400A 0C00 - 0x400A 0FFF	TZBMPC1
		0x500A 0800 - 0x500A 0BFF	0x400A 0800 - 0x400A 0BFF	TZBMPC0
		0x500A 0400 - 0x500A 07FF	0x400A 0400 - 0x400A 07FF	TZIAC
		0x500A 0000 - 0x500A 03FF	0x400A 0000 - 0x400A 03FF	TZSPC
		0x5008 0400 - 0x5009 FFFF	0x4008 0400 - 0x4009 FFFF	Reserved
		0x5008 0000 - 0x5008 03FF	0x4008 0000 - 0x4008 03FF	ICACHE
		0x5003 3000 - 0x5007 FFFF	0x4003 3000 - 0x4007 FFFF	Reserved

Pre-defined Regions	Bus	Secure boundary address	Non-Secure boundary address	Peripherals
		0x5003 0000 - 0x5003 2FFF	0x4003 0000 - 0x4003 2FFF	Wi-Fi
		0x5002 BC00 - 0x5002 FFFF	0x4002 BC00 - 0x4002 FFFF	Reserved
		0x5002 B000 - 0x5002 BBFF	0x4002 B000 - 0x4002 BBFF	Reserved
		0x5002 A000 - 0x5002 AFFF	0x4002 A000 - 0x4002 AFFF	Reserved
		0x5002 8000 - 0x5002 9FFF	0x4002 8000 - 0x4002 9FFF	Reserved
		0x5002 6800 - 0x5002 7FFF	0x4002 6800 - 0x4002 7FFF	Reserved
		0x5002 6400 - 0x5002 67FF	0x4002 6400 - 0x4002 67FF	DMA1
		0x5002 6000 - 0x5002 63FF	0x4002 6000 - 0x4002 63FF	DMA0
		0x5002 5C00 - 0x5002 5FFF	0x4002 5C00 - 0x4002 5FFF	Reserved
		0x5002 5800 - 0x5002 5BFF	0x4002 5800 - 0x4002 5BFF	QSPI_FLASH(REG)
		0x5002 5400 - 0x5002 57FF	0x4002 5400 - 0x4002 57FF	SQPI_PSRAM(REG)
		0x5002 5000 - 0x5002 53FF	0x4002 5000 - 0x4002 53FF	Reserved
		0x5002 4000 - 0x5002 4FFF	0x4002 4000 - 0x4002 4FFF	TSI
		0x5002 3C00 - 0x5002 3FFF	0x4002 3C00 - 0x4002 3FFF	Reserved
		0x5002 3800 - 0x5002 3BFF	0x4002 3800 - 0x4002 3BFF	RCU
		0x5002 3400 - 0x5002 37FF	0x4002 3400 - 0x4002 37FF	Reserved
		0x5002 3000 - 0x5002 33FF	0x4002 3000 - 0x4002 33FF	CRC
		0x5002 2C00 - 0x5002 2FFF	0x4002 2C00 - 0x4002 2FFF	Reserved
		0x5002 2800 - 0x5002 2BFF	0x4002 2800 - 0x4002 2BFF	EFUSE
		0x5002 2400 - 0x5002 27FF	0x4002 2400 - 0x4002 27FF	Reserved
		0x5002 2000 - 0x5002 23FF	0x4002 2000 - 0x4002 23FF	FMC
		0x5002 1C00 - 0x5002 1FFF	0x4002 1C00 - 0x4002 1FFF	Reserved
		0x5002 1800 - 0x5002 1BFF	0x4002 1800 - 0x4002 1BFF	Reserved
		0x5002 1400 - 0x5002 17FF	0x4002 1400 - 0x4002 17FF	Reserved
		0x5002 1000 - 0x5002 13FF	0x4002 1000 - 0x4002 13FF	Reserved
		0x5002 0C00 - 0x5002 0FFF	0x4002 0C00 - 0x4002 0FFF	Reserved
		0x5002 0800 - 0x5002 0BFF	0x4002 0800 - 0x4002 0BFF	GPIOC
		0x5002 0400 - 0x5002 07FF	0x4002 0400 - 0x4002 07FF	GPIOB
		0x5002 0000 - 0x5002 03FF	0x4002 0000 - 0x4002 03FF	GPIOA
APB2		0x5001 8800 - 0x5001 FFFF	0x4001 8800 - 0x4001 FFFF	Reserved
		0x5001 8400 - 0x5001 87FF	0x4001 8400 - 0x4001 87FF	TIMER16
		0x5001 8000 - 0x5001 83FF	0x4001 8000 - 0x4001 83FF	TIMER15
		0x5001 7C00 - 0x5001 7FFF	0x4001 7C00 - 0x4001 7FFF	Reserved
		0x5001 7800 - 0x5001 7BFF	0x4001 7800 - 0x4001 7BFF	Wi-Fi_RF
		0x5001 6800 - 0x5001 77FF	0x4001 6800 - 0x4001 77FF	Reserved
		0x5001 6000 - 0x5001 67FF	0x4001 6000 - 0x4001 67FF	HPDF
		0x5001 5800 - 0x5001 5FFF	0x4001 5800 - 0x4001 5FFF	Reserved
		0x5001 5400 - 0x5001 57FF	0x4001 5400 - 0x4001 57FF	Reserved
		0x5001 4C00 - 0x5001 53FF	0x4001 4C00 - 0x4001 53FF	Reserved
		0x5001 4800 - 0x5001 4BFF	0x4001 4800 - 0x4001 4BFF	Reserved

Pre-defined Regions	Bus	Secure boundary address	Non-Secure boundary address	Peripherals
APB1		0x5001 4400 - 0x5001 47FF	0x4001 4400 - 0x4001 47FF	Reserved
		0x5001 4000 - 0x5001 43FF	0x4001 4000 - 0x4001 43FF	Reserved
		0x5001 3C00 - 0x5001 3FFF	0x4001 3C00 - 0x4001 3FFF	EXTI
		0x5001 3800 - 0x5001 3BFF	0x4001 3800 - 0x4001 3BFF	SYSCFG
		0x5001 3400 - 0x5001 37FF	0x4001 3400 - 0x4001 37FF	Reserved
		0x5001 3000 - 0x5001 33FF	0x4001 3000 - 0x4001 33FF	SPI0
		0x5001 2C00 - 0x5001 2FFF	0x4001 2C00 - 0x4001 2FFF	SDIO
		0x5001 2400 - 0x5001 2BFF	0x4001 2400 - 0x4001 2BFF	Reserved
		0x5001 2000 - 0x5001 23FF	0x4001 2000 - 0x4001 23FF	ADC
		0x5001 1400 - 0x5001 1FFF	0x4001 1400 - 0x4001 1FFF	Reserved
		0x5001 1000 - 0x5001 13FF	0x4001 1000 - 0x4001 13FF	USART2
		0x5001 0800 - 0x5001 0FFF	0x4001 0800 - 0x4001 0FFF	Reserved
		0x5001 0400 - 0x5001 07FF	0x4001 0400 - 0x4001 07FF	Reserved
		0x5001 0000 - 0x5001 03FF	0x4001 0000 - 0x4001 03FF	TIMER0
		0x5000 7400 - 0x5000 FFFF	0x4000 D000 - 0x4000 FFFF	Reserved
APB2		0x5000 CC00 - 0x5000 CFFF	0x4000 CC00 - 0x4000 CFFF	Reserved
		0x5000 7400 - 0x5000 CBFF	0x4000 7400 - 0x4000 CBFF	Reserved
		0x5000 7000 - 0x5000 73FF	0x4000 7000 - 0x4000 73FF	PMU
		0x5000 6C00 - 0x5000 6FFF	0x4000 6C00 - 0x4000 6FFF	Reserved
		0x5000 5C00 - 0x5000 6BFF	0x4000 5C00 - 0x4000 6BFF	Reserved
		0x5000 5800 - 0x5000 5BFF	0x4000 5800 - 0x4000 5BFF	I2C1
		0x5000 5400 - 0x5000 57FF	0x4000 5400 - 0x4000 57FF	I2C0
		0x5000 4C00 - 0x5000 53FF	0x4000 4C00 - 0x4000 53FF	Reserved
		0x5000 4800 - 0x5000 4BFF	0x4000 4800 - 0x4000 4BFF	USART0
		0x5000 4400 - 0x5000 47FF	0x4000 4400 - 0x4000 47FF	USART1
		0x5000 4000 - 0x5000 43FF	0x4000 4000 - 0x4000 43FF	Reserved
		0x5000 3C00 - 0x5000 3FFF	0x4000 3C00 - 0x4000 3FFF	Reserved
		0x5000 3800 - 0x5000 3BFF	0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x5000 3400 - 0x5000 37FF	0x4000 3400 - 0x4000 37FF	I2S1_add
		0x5000 3000 - 0x5000 33FF	0x4000 3000 - 0x4000 33FF	FWDGT
		0x5000 2C00 - 0x5000 2FFF	0x4000 2C00 - 0x4000 2FFF	WWDT
		0x5000 2800 - 0x5000 2BFF	0x4000 2800 - 0x4000 2BFF	RTC
		0x5000 2400 - 0x5000 27FF	0x4000 2400 - 0x4000 27FF	Reserved
		0x5000 2000 - 0x5000 23FF	0x4000 2000 - 0x4000 23FF	Reserved
		0x5000 1C00 - 0x5000 1FFF	0x4000 1C00 - 0x4000 1FFF	Reserved
		0x5000 1800 - 0x5000 1BFF	0x4000 1800 - 0x4000 1BFF	Reserved
		0x5000 1400 - 0x5000 17FF	0x4000 1400 - 0x4000 17FF	Reserved
		0x5000 1000 - 0x5000 13FF	0x4000 1000 - 0x4000 13FF	TIMER5
		0x5000 0C00 - 0x5000 0FFF	0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x5000 0800 - 0x5000 0BFF	0x4000 0800 - 0x4000 0BFF	TIMER3

Pre-defined Regions	Bus	Secure boundary address	Non-Secure boundary address	Peripherals
SRAM	AHB	0x5000 0400 - 0x5000 07FF	0x4000 0400 - 0x4000 07FF	TIMER2
		0x5000 0000 - 0x5000 03FF	0x4000 0000 - 0x4000 03FF	TIMER1
		0x3007 0000 - 0x3FFF FFFF	0x2007 0000 - 0x2FFF FFFF	Reserved
		0x3004 0000 - 0x2006 FFFF	0x2004 0000 - 0x2006 FFFF	SRAM3 (192KB)
		0x3002 0000 - 0x3003 FFFF	0x2002 0000 - 0x2003 FFFF	SRAM2 (128KB)
		0x3001 0000 - 0x3001 FFFF	0x2001 0000 - 0x2001 FFFF	SRAM1 (64KB)
	Code	0x3000 0000 - 0x3000 FFFF	0x2000 0000 - 0x2000 FFFF	SRAM0 (64KB)
		-	0x1000 0000 - 0x1FFF FFFF	External memories remap
		-	0x0BFF 8000 - 0x0BFF FFFF	Reserved
		0x0FF8 8000 - 0x0FFF FFFF	0x0BF8 0000 - 0x0BFF 7FFF	Reserved
		0x0FF8 4000 - 0x0FF8 7FFF	-	ROM(16KB)
		0x0FF8 0000 - 0x0FF8 3FFF	-	GSSA(16KB)
		0x0FF4 E000 - 0x0FF7 FFFF	0x0BF4 E000 - 0x0BF7 FFFF	ROM(200KB)
		-	0x0BF4 6000 - 0x0BF4 CFFF	Reserved
		-	0x0BF4 0000 - 0x0BF4 5FFF	ROM(24KB)
		0x0E07 0000 - 0x0FF4 DFFF	0x0A07 0000 - 0x0BF3 FFFF	Reserved
		0x0E04 0000 - 0x0E06 FFFF	0x0A04 0000 - 0x0A06 FFFF	SRAM3 (192KB)
		0x0E02 0000 - 0x0E03 FFFF	0x0A02 0000 - 0x0A03 FFFF	SRAM2 (128KB)
		0x0E01 0000 - 0x0E01 FFFF	0x0A01 0000 - 0x0A01 FFFF	SRAM1 (64KB)
		0x0E00 0000 - 0x0E00 FFFF	0x0A00 0000 - 0x0A00 FFFF	SRAM0 (64KB)
		0x0C20 0000 - 0x0DFF FFFF	0x0820 0000 - 0x09FF FFFF	Reserved
		0x0C00 0000 - 0x0C1F FFFF	0x0800 0000 - 0x081F FFFF	Flash memory
		-	0x0000 0000 - 0x07FF FFFF	External memories remap

1.4.1. On-chip SRAM memory

The GD32W51x series of devices contain up to 448 KB of on-chip SRAM (SRAM0 64 KB, SRAM1 64 KB, SRAM2 128 KB, SRAM3 192 KB) which starts at: the non-secure address 0x2000 0000 and the secure address 0x3000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses.

When the TrustZone security is enabled, all SRAMs are secure after reset. The SRAM can be programmed as non-secure with a block granularity, using TZBMPC (TrustZone block-based memory protection controller) in TZPCU controller. The granularity of SRAM secure/non-secure block-based is a page of 256 bytes.

1.4.2. SRAM1 Write protection

The SRAM1 can be write protected with a page granularity of 1 Kbyte.

The write protection can be enabled in SYSCFG SRAM1 write protection register (SYSCFG_SWPRx (x=0, 1)) in the SYSCFG block. This is a register with write ‘1’ once

mechanism, which means by writing '1' on a bit it will setup the write protection for that page of SRAM and it can be removed/cleared by a system reset only.

1.4.3. SRAM1 security protection

The SRAM1 is protected with the security protection (SPC).

1.4.4. SRAM1 Erase

The SRAM1 erase can also be requested by software by setting the bit SRAM1ERS in the SYSCFG SRAM1 control and status register (SYSCFG_SCS).

The SRAM1 can be erased with a system reset using the option bit SRAM1_RST in the user option byte.

The SRAM1 is also erased by a Backup domain reset, any tamper detection or setting the BKERASE bit in the RTC_TAMP register.

1.4.5. On-chip flash memory overview

The devices provide up to 2048 KB of on-chip flash memory and the flash memory organized into 512 pages with 4 KB and 256KB information block for the boot loader.

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

1.5. Boot configuration

At startup, a BOOT0 pin, a BOOT1 pin are used to select the boot memory address.

The BOOT0 value may come from the BOOT0 pin or from the value of SWBOOT0 bit in the EFUSE_CTL register to free the GPIO pad if needed.

The BOOT1 value may come from the PA14 pin or from the value of SWBOOT1 bit in the EFUSE_CTL register to free the GPIO pad if needed.

Table 1-6. BOOT0 modes

SWBOOT0	EFBOOT0	BOOT0 PC8 pin	BOOT0
0	-	0	0
0	-	1	1
1	0	-	0
1	1	-	1

Table 1-7. BOOT1 modes

SWBOOT1	EFBOOT1	BOOT1 PA14 pin	BOOT1
0	-	0	0
0	-	1	1
1	0	-	0
1	1	-	1

Refer to [Table 1-8. Boot address modes when TrustZone is disabled \(TZEN=0\)](#) and [Table 1-9. Boot modes when TrustZone is enabled \(TZEN=1\)](#) for boot address when TrustZone is disabled and enabled respectively. When the EFBOOTLK bit in the EFUSE_CTL register is set, the boot memory address selected according to boot1 and boot0.

Table 1-8. Boot address modes when TrustZone is disabled (TZEN=0)

EFBOOTLK	BOOT0	BOOT1	Boot address	Boot area
0	0	-	0x08000000	SIP Flash when cfg_qspi is 0 QSPI Flash when cfg_qspi is 1
0	1	0	0x0BF40000	Bootloader / ROM
0	1	1	0x0A000000	SRAM0
1	0	-	0x08000000	SIP Flash when cfg_qspi is 0 QSPI Flash when cfg_qspi is 1
1	1	-	0x0BF40000	Bootloader / ROM

When TrustZone is enabled by setting the TZEN option bit, the boot space must be in secure area.

Table 1-9. Boot modes when TrustZone is enabled (TZEN=1)

GSSAC MD == 8'hc ⁽¹⁾	EFBOOTLK	BOOT0	BOOT1	EFSB	Boot address	Boot area
0	0	0	-	0	0x0C000 000	SPI Flash when cfg_qspi is 0 QSPI Flash when cfg_qspi is 1
0	0	0	-	1	0X0FF84 000	secure boot
0	0	1	0	-	0x0FF80 000	GSSA
0	0	1	1	-	0x0E000 000	SRAM0

GSSAC MD == 8'hc⁽¹⁾	EFBOOTLK	BOOT0	BOOT1	EFSB	Boot address	Boot area
-	1	0	-	0	0x0C000 000	SPI Flash when cfg_qspi is 0 QSPI Flash when cfg_qspi is 1
-	1	0	-	1	0X0FF84 000	secure boot
-	1	1	-	-	0x0FF80 000	GSSA
1	0	-	-	-	0x0FF80 000	GSSA

Note: (1) When the GSSACMD bitfield is 0x0C, it means 1, otherwise it means 0.

The BOOTx (x=0/1) value (either coming from the pin or the EFBOOTx bit) is latched upon reset release. It is up to the user to set BOOTx values to select the required boot mode. The BOOTx pin or EFBOOTx bit (depending on the EFBOOTLK and SWBOOTx bit value in the EFUSE_CTL register) is also re-sampled when exiting from Standby mode. Consequently, they must be kept in the required Boot mode configuration in Standby mode. After startup delay, the selection of the boot area is done before releasing the processor reset.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. The boot loader can be activated through one of the following serial interfaces: USART0 (PA8, PB15), USART1 (PA2, PA3) and USART2(PB10, PB11).

1.6. System configuration controller(SYSCFG)

1.6.1. SYSCFG main features

- Setting SRMA1 write protection and software erase
- Configuring FPU interrupts
- Configuring TrustZone security register access

1.6.2. SYSCFG TrustZone security and privilege

When the TrustZone security is activated, the SYSCFG is able to secure registers from being modified by non-secure accesses. The TrustZone security is activated by the TZEN option bit in the EFUSE_TZCTL register or the TZEN option bit in the option byte. A non-secure read/write access to a secured register is RAZ/WI and generates an illegal access event. An illegal access interrupt is generated if the SYSCFG illegal access event is enabled in the TZIAC_INTEN register.

1.6.3. SYSCFG registers

SYSCFG secure access base address: 0x5001 3800

SYSCFG son-secure access base address: 0x4001 3800

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 BOOT0 and BOOT1 pins)

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
															BOOT_MODE[1:0]

r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	BOOT_MODE[1:0]	Boot mode (Refer to Boot configuration for details) Bit0 is mapping to the BOOT0 value; the value of bit1 is the opposite of the BOOT1_n option bit value. x0: Boot from the others 01: Refer to Boot configuration for details 11: Boot from the embedded SRAM

EXTI sources selection register 0 (SYSCFG_EXTISS0)

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
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15:12	EXTI3_SS[3:0]	EXTI 3 sources selection 0000: PA3 pin 0001: PB3 pin 0010: PC3 pin Other configurations are reserved.
11:8	EXTI2_SS[3:0]	EXTI 2 sources selection 0000: PA2 pin 0001: PB2 pin 0010: PC2 pin Other configurations are reserved.
7:4	EXTI1_SS[3:0]	EXTI 1 sources selection 0000: PA1 pin 0001: PB1 pin 0010: PC1 pin Other configurations are reserved.
3:0	EXTI0_SS[3:0]	EXTI 0 sources selection 0000: PA0 pin 0001: PB0 pin 0010: PC0 pin Other configurations are reserved.

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]	EXTI 7 sources selection 0000: PA7 pin 0001: PB7 pin 0010: PC7 pin Other configurations are reserved.

11:8	EXTI6_SS[3:0]	EXTI 6 sources selection 0000: PA6 pin 0001: PB6 pin 0010: PC6 pin Other configurations are reserved.
7:4	EXTI5_SS[3:0]	EXTI 5 sources selection 0000: PA5 pin 0001: PB5 pin 0010: PC5 pin Other configurations are reserved.
3:0	EXTI4_SS[3:0]	EXTI 4 sources selection 0000: PA4 pin 0001: PB4 pin 0010: PC4 pin Other configurations are reserved.

EXTI sources selection register 2 (SYSCFG_EXTISSL2)

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															
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]	EXTI 11 sources selection 0000: PA11 pin 0001: PB11 pin Other configurations are reserved.
11:8	EXTI10_SS[3:0]	EXTI 10 sources selection 0000: PA10 pin 0001: PB10 pin Other configurations are reserved.
7:4	EXTI9_SS[3:0]	EXTI 9 sources selection 0000: PA9 pin

Other configurations are reserved.

3:0	EXTI8_SS[3:0]	EXTI 8 sources selection
	0000:	PA8 pin
	0001:	PB8 pin
	0010:	PC8 pin
		Other configurations are reserved.

EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

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

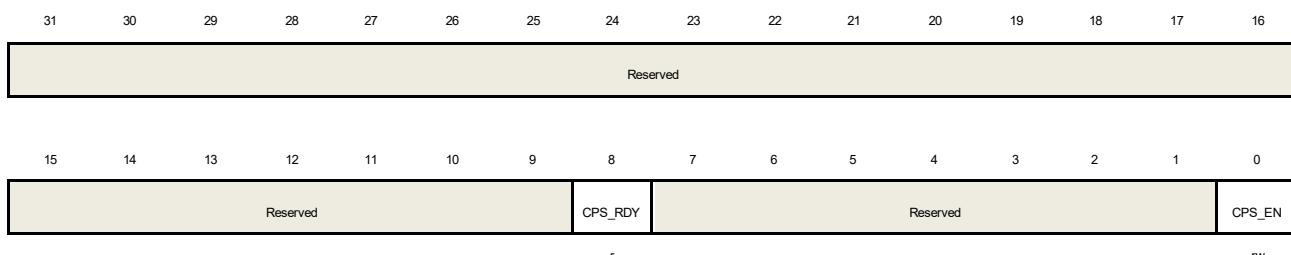
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection 0000: PA15 pin 0001: PB15 pin Other configurations are reserved.
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection 0000: PA14 pin 0001: PB14 pin Other configurations are reserved.
7:4	EXTI13_SS[3:0]	EXTI 13 sources selection 0000: PA13 pin 0001: PB13 pin Other configurations are reserved.
3:0	EXTI12_SS[3:0]	EXTI 12 sources selection 0000: PA12 pin 0001: PB12 pin Other configurations are reserved.

I/O compensation control register (SYSCFG_CPSCTL)

Address offset: 0x20

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	CPS_RDY	Compensation cell ready flag 0: I/O compensation cell not ready 1: I/O compensation cell ready
7:1	Reserved	Must be kept at reset value
0	CPS_EN	Compensation cell power-down 0: I/O compensation cell power-down mode 1: I/O compensation cell enabled

SYSCFG secure configuration register (SYSCFG_SECFG)

Address offset: 0x40

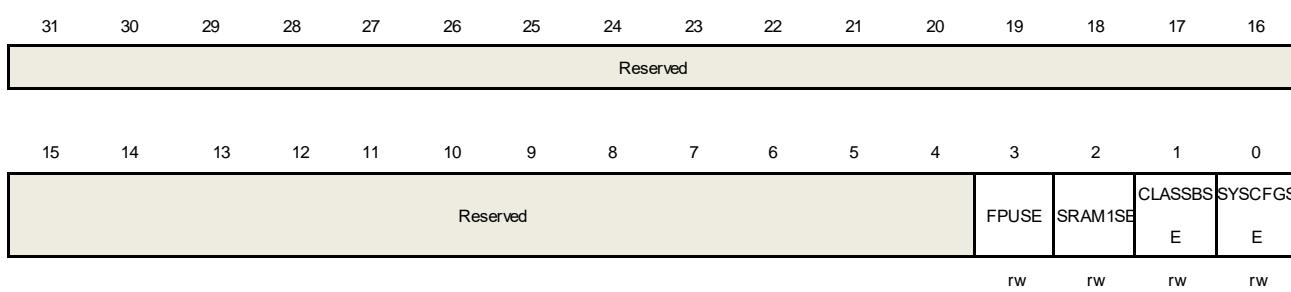
Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register provides write and read access security only when the access is secure. A non-secure write or read access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), this register is RAZ/WI.

This register can be read and written by privileged and unprivileged access.

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



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	FPUSE	FPU security 0: SYSCFG_FPUINTEN register can be written by secure and non-secure access 1: SYSCFG_FPUINTEN register can be written by secure access only.
2	SRAM1SE	SRAM1 security 0: SYSCFG_SKEY, SYSCFG_SCS and SYSCFG_SWPx registers can be written by secure and non-secure access 1: SYSCFG_SKEY, SYSCFG_SCS and SYSCFG_SWPx register can be written by secure access only.
1	CLASSBSE	ClassB security. 0: SYSCFG_CFG1 register can be written by secure and non-secure access 1: SYSCFG_CFG1 register can be written by secure access only.
0	SYSCFGSE	SYSCFG clock control security 0: SYSCFG configuration clock in RCU registers can be written by secure and non-secure access 1: SYSCFG configuration clock in RCU registers can be written by secure access only.

FPU interrupt enable register (SYSCFG_FPUINTEN)

Address offset: 0x48

Reset value: 0x0000 001F

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the FPUSE bit in the SYSCFG_SECFG register. When FPUSE bit is 0, there is no access restriction.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged access only. Unprivileged access is RAZ/WI.

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										IXIE	IDIE	OVFIE	UFIE	DZIE	IOPIE
rw										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
-------------	---------------	---------------------

31:6	Reserved	Must be kept at reset value.
5	IXIE	Inexact interrupt enable bit 0: Inexact interrupt disable. 1: Inexact interrupt enable.
4	IDIE	Input denormal interrupt enable bit 0: Input denormal interrupt disable. 1: Input denormal interrupt enable.
3	OVFIE	Overflow interrupt enable bit 0: Overflow interrupt disable. 1: Overflow interrupt enable.
2	UFIE	Underflow interrupt enable bit 0: Underflow interrupt disable. 1: Underflow interrupt enable.
1	DZIE	Divide by 0 interrupt enable bit 0: Divide by 0 interrupt disable. 1: Divide by 0 interrupt enable.
0	IOPIZ	Invalid operation interrupt enable bit 0: Invalid operation interrupt disable. 1: Invalid operation interrupt enable.

SYSCFG CPU non-secure lock register (SYSCFG_CNSLOCK)

Address offset: 0x4C

Reset value: 0x0000 0000

This register is used to lock the configuration of non-secure MPU and VTOR_NS registers. When the system is secure (TZEN = 1), read/write access is no access restriction.

When the system is not secure (TZEN=0), this register is RAZ/WI.

This register can be read and written by privileged access only. Unprivileged access is RAZ/WI.

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												LOCKNS MPU	LOCKNSV TOR	rs	rs

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	LOCKNSMPU	<p>Non-secure MPU registers lock</p> <p>This bit is set by software and cleared only by a system reset. When set, it disables write access to non-secure MPU_CTRL_NS, MPU_RNR_NS and MPU_RBAR_NS registers.</p> <p>0: Non-secure MPU registers write is enabled 1: Non-secure MPU registers write is disabled</p>
0	LOCKNSVTOR	<p>VTOR_NS register lock</p> <p>This bit is set by software and cleared only by a system reset.</p> <p>0: VTOR_NS register write is enabled 1: VTOR_NS register write is disabled</p>

SYSCFG CPU secure lock register (SYSCFG_CSLOCK)

Address offset: 0x50

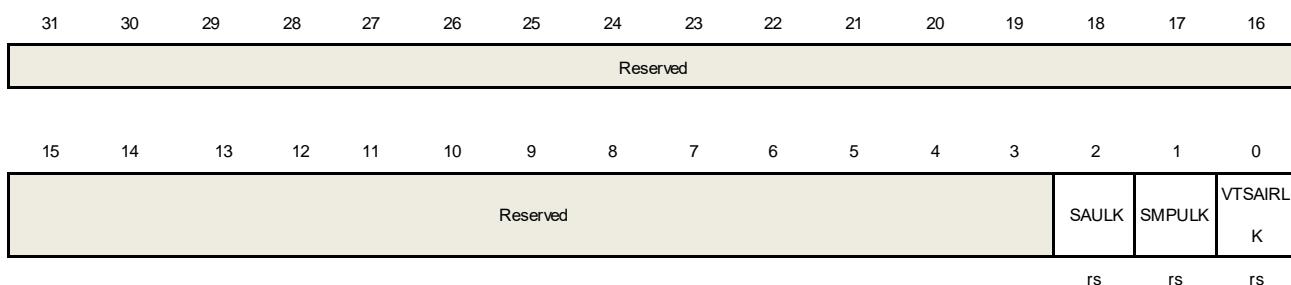
Reset value: 0x0000 0000

This register is used to lock the configuration of PRIS and BFHFNMINS bits in the AIRCR register, SAU, secure MPU and VTOR_S registers. When the system is secure (TZEN =1), this register can be written only when the access is secure. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), this register is RAZ/WI

This register can be read and written by privileged access only. Unprivileged access is RAZ/WI.

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



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	SAULK	<p>SAU registers lock.</p> <p>This bit is set by software and cleared only by a system reset. When set, it disables write access to SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers.</p> <p>0: SAU registers write is enabled</p>

		1: SAU registers write is disabled
1	SMPULK	<p>Secure MPU registers lock.</p> <p>This bit is set by software and cleared only by a system reset. When is set, it disables write access to secure MPU_CTRL, MPU_RNR and MPU_RBAR registers.</p>
		0: Secure MPU registers writes is enabled
		1: Secure MPU registers writes is disabled.
0	VTSAIRLK	<p>VTOR_S register and AIRCR register bits lock.</p> <p>This bit is set by software and cleared only by a system reset. When is set, it disables write access to VTOR_S register, PRIS and BFHFNMINS bits in the AIRCR register.</p>
		0: VTOR_S register PRIS and BFHFNMINS bits in the AIRCR register write is enabled
		1: VTOR_S register PRIS and BFHFNMINS bits in the AIRCR register write is disabled

SYSCFG configuration register 1 (SYSCFG_CFG1)

Address offset: 0x54

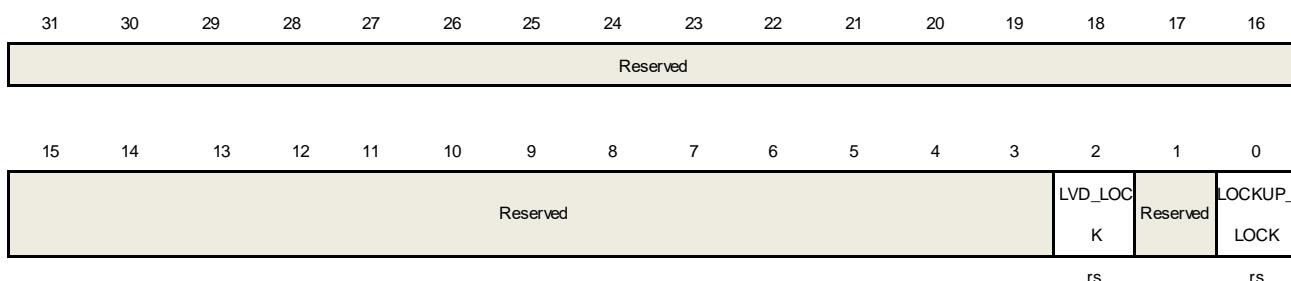
Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the CLASSBSE bit in the SYSCFG_SECFG register. When CLASSBSE bit is set, only secure access is allowed. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged and unprivileged access.

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



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	LVD_LOCK	<p>LVD lock enable bit.</p> <p>This bit is set by software and cleared only by a system reset. It can be used to enable and lock the LVD connection to TIMER0/15/16 Break input, as well as the</p>

LVDEN and LVDT[2:0] in the PMU_CTL0 register.
 0: LVD interrupt disconnected from TIMER0/15/16 Break input. LVDEN and LVDT[2:0] bits can be programmed by the application
 1: LVD interrupt connected to TIMER0/15/16 Break input, LVDEN and LVDT[2:0] bits are read only

1	Reserved	Must be kept at reset value.
0	LOCKUP_LOCK	<p>Cortex®-M33 LOCKUP (hardfault) output enable bit. This bit is set by software and cleared only by a system reset. It can be used to enable and lock the connection of Cortex®-M33 LOCKUP (hardfault) output to TIMER0/15/16 Break input.</p> <p>0: Cortex®-M33 LOCKUP output disconnected from TIMER0/15/16 Break inputs. 1: Cortex®-M33 LOCKUP output connected to TIMER0/15/16 Break inputs.</p>

SYSCFG SRAM1 control and status register (SYSCFG_SCS)

Address offset: 0x58

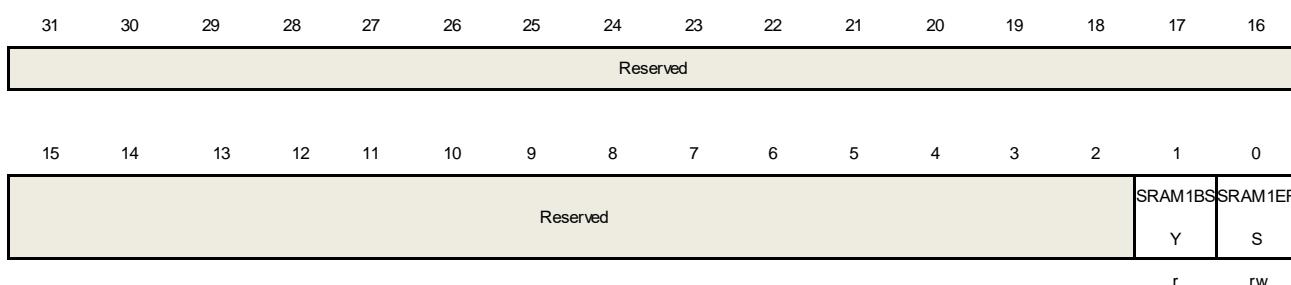
Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the SRAM1SE bit in the SYSCFG_SECFG register. When SRAM1SE bit is set, only secure access is allowed. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged and unprivileged access.

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SRAM1BSY	<p>SRAM1 busy by erase operation.</p> <p>0: No SRAM1 erase operation is ongoing</p> <p>1: SRAM1 erase operation is ongoing.</p>
0	SRAM1ERS	<p>SRAM1 erase</p> <p>Setting this bit starts a hardware SRAM1 erase operation. This bit is automatically</p>

cleared at the end of the SRAM1 erase operation

Note: This bit is write-protected: setting this bit is possible only after the correct key sequence is written in the SYSCFG_SKEY register.

SYSCFG SRAM1 key register (SYSCFG_SKEY)

Address offset: 0x5C

Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the SRAM1SE bit in the SYSCFG_SECFG register. When SRAM1SE bit is set, only secure access is allowed. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged and unprivileged access.

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								KEY[7:0]							
W															

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	KEY[7:0]	<p>SRAM1 write protection key for software erase.</p> <p>The following steps are required to unlock the write protection of the SRAM1ERS bit in the SYSCFG_SCS register.</p> <ol style="list-style-type: none"> 1. Write "0xCA" into Key[7:0] 2. Write "0x53" into Key[7:0] <p>Note: Writing a wrong key reactivates the write protection.</p>

SYSCFG SRAM1 write protection register 0 (SYSCFG_SWP0)

Address offset: 0x60

Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the SRAM1SE bit in the SYSCFG_SECFG register. When SRAM1SE bit is set, only secure access is allowed. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged and unprivileged access.

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
P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
rs															

Bits	Fields	Descriptions
31:0	PxWP (x = 0 to 31)	SRAM1 1 Kbyte page x write protection. These bits are set by software and cleared only by a system reset. 0: Write protection of SRAM1 1 Kbyte page x is disabled 1: Write protection of SRAM1 1 Kbyte page x is enabled.

SYSCFG SRAM1 write protection register 1 (SYSCFG_SWP1)

Address offset: 0x64

Reset value: 0x0000 0000

When the system is secure (TZEN =1), this register can be protected against non-secure access by setting the SRAM1SEC bit in the SYSCFG_SECCFGR register. When SRAM1SEC bit is set, only secure access is allowed. A non-secure read/write access is RAZ/WI and generates an illegal access event.

When the system is not secure (TZEN=0), there is no access restriction.

This register can be read and written by privileged and unprivileged access.

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
P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
rs															

Bits	Fields	Descriptions
31:0	PxWP (x = 32 to 63)	SRAM1 1 Kbyte page x write protection. These bits are set by software and cleared only by a system reset. 0: Write protection of SRAM1 1 Kbyte page x is disabled 1: Write protection of SRAM1 1 Kbyte page x is enabled.

SYSCFG_GSSA command register (SYSCFG_GSSACMD)

Address offset: 0x6C

Reset value: 0x0000 0000, reset by power reset only, system reset has no affected.

When the system is secure (TZEN =1), this register can be read and written only when the APB access is secure. Otherwise it is RAZ/WI.

When the system is not secure (TZEN=0), this register is RAZ/WI.

This register can be read and written by privileged and unprivileged access.

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								GSSACMD							
rw															

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	GSSACMD	GSSA commands Defines a command to be executed by the GSSA.

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.

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

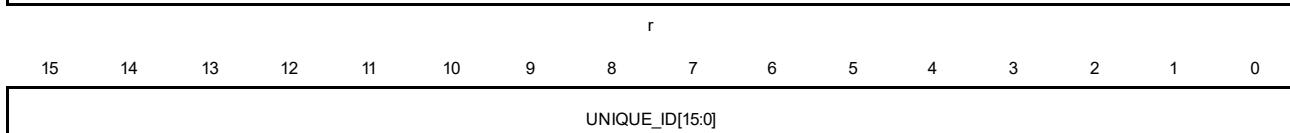
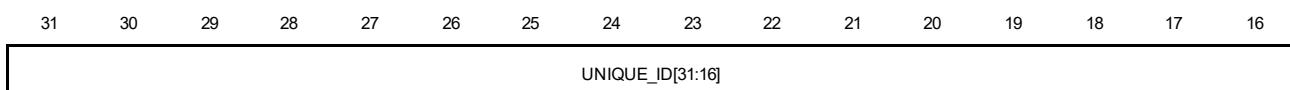
Bits	Fields	Descriptions
------	--------	--------------

31:16	SRAM_DENSITY	SRAM density
	[15:0]	The value indicates the on-chip SRAM density of the device in Kbytes.
		Example: 0x0008 indicates 8 Kbytes.
15:0	FLASH_DENSITY	Flash memory density
	[15:0]	The value indicates the Flash memory density of the device in Kbytes.
		Example: 0x0020 indicates 32 Kbytes.

1.7.2. Unique device ID (96 bits)

Base address: 0x1FFF F7E8

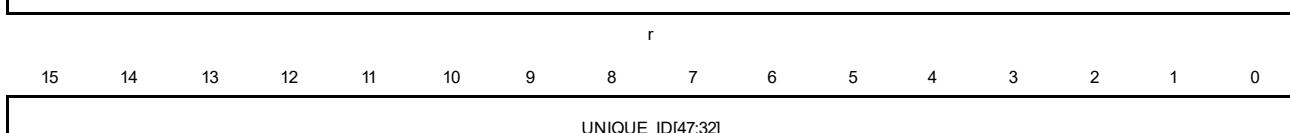
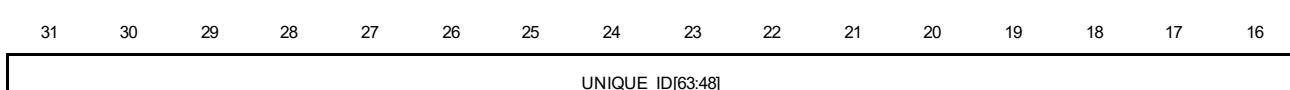
The value is factory programmed and can never be altered by user.



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

Base address: 0x1FFF F7EC

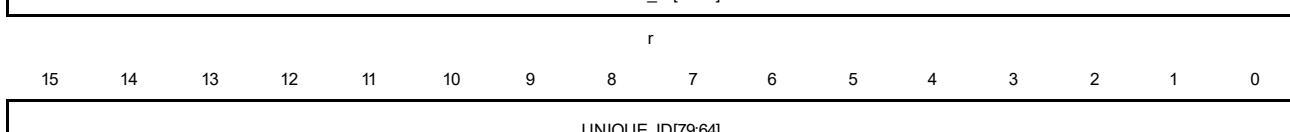
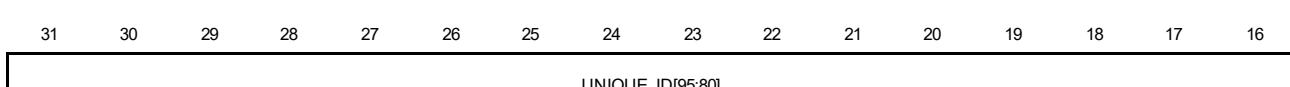
The value is factory programmed and can never be altered by user.



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.



r

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

2. Flash memory controller (FMC)

2.1. Overview

For GD32W51x, the flash memory architecture is divided into FMC mode and QSPI mode. For FMC mode, there is on-chip (SIP, system in package) flash. For QSPI mode, there is EXT flash. The flash memory controller, FMC, provides all the necessary functions for the on-chip Flash Memory. There also provide page erase, mass erase, and word program for flash memory.

2.2. Characteristics

- Two memory organizations:
 - Main :
 - FMC mode: SIP Flash (max: 2MB)
 - QSPI mode: EXT Flash (max: 32MB)
 - Information block : 256KB bootloader
- In FMC mode, the on-chip Flash pagesize is 4KB.
- In QSPI mode, read by FMC or QSPI interface and program/erase by QSPI interface.
- Supports RTDEC (decrypt by AES real-time) fuction.
- Supports program and read EFUSE (store AES key and initial registers).
- Word programming, page erase and mass erase operation for FMC mode.
- Security protection for FMC mode/QSPI mode to prevent illegal code/data access.
- Page erase/program protection to prevent unexpected operation for FMC mode.

2.3. System overview

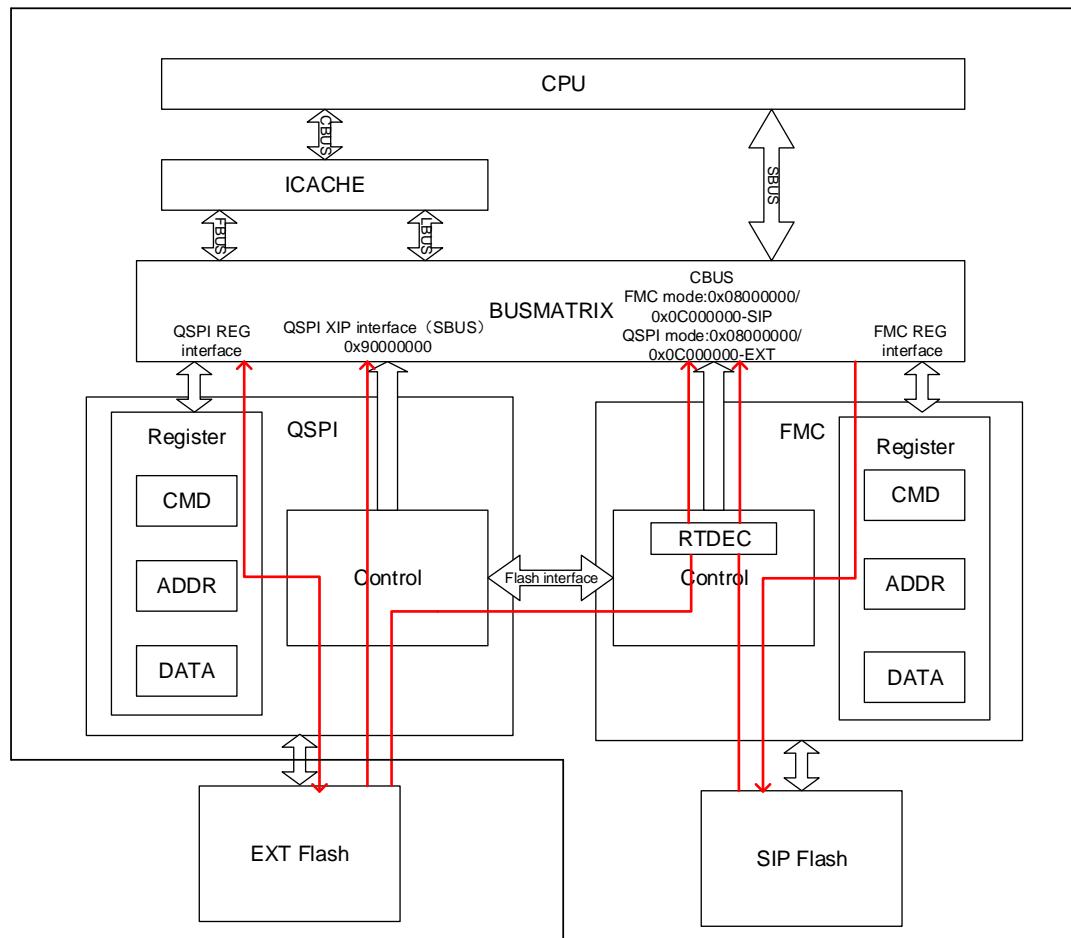
The figure [Figure 2-1. FMC bus in GD32W51x](#) shows the location of FMC bus in the system.

In FMC mode, FMC supports CBUS interface (read-only, RTDEC), FMC registers (write-only). In other words, SIP Flash can be read by CBUS interface with RTDEC fuction and write by FMC REG interface.

In QSPI mode, QSPI supports QSPI REG interface (read-write), QSPI XIP interface (read-only, no RTDEC) and CBUS interface (read-only, 0x0800 0000/0x0C00 0000, RTDEC). In other words, EXT Flash can be read and write by QSPI REG interface, read by QSPI XIP interface and read by CBUS interface with RTDEC fuction.

Note: please refer to [Read operations](#) for RTDEC function.

Figure 2-1. FMC bus in GD32W51x



2.4. Function overview

2.4.1. Flash memory architecture

For FMC mode, the flash memory consists of 2MB main flash organized into 512 pages with 4 KB and 256KB information block for the boot loader. Each page can be erased individually. The structure of EXT flash depends on the specifics of the external flash. The following table shows the details of flash organization in FMC mode.

Table 2-1. GD32W51x base address and size for flash memory (FMC mode)

Block	Name	Address range	size(bytes)
Main flash block (SIP Flash)	Page 0	0x0800 0000 - 0x0800 0FFF ⁽¹⁾	4KB
		0x0C00 0000 - 0x0C00 0FFF ⁽²⁾	
	Page 1	0x0800 1000 - 0x0800 1FFF ⁽¹⁾	4KB
		0x0C00 1000 - 0x0C00 1FFF ⁽²⁾	
	Page 2	0x0800 2000 - 0x0800 2FFF ⁽¹⁾	4KB
		0x0C00 2000 - 0x0C00 2FFF ⁽²⁾	

Block	Name	Address range	size(bytes)
	Page 511	0x081F F000 - 0x081F FFFF ⁽¹⁾ 0x0C1F F000 - 0x0C1F FFFF ⁽²⁾	4KB
Bootloader	normal ⁽³⁾	0x0BF4 0000 - 0x0BF4 5FFF	24KB
	secure boot ⁽⁴⁾	0x0FF8 4000 - 0x0FF8 7FFF	16KB
	secure region2 ⁽⁵⁾	0x0FF4 E000 - 0x0FF7 FFFF	200KB
	GSSA ⁽⁴⁾	0x0FF8 0000 - 0x0FF8 3FFF	16KB

In QSPI mode, the structure of EXT flash depends on the specifics of the external flash. The following table shows the details of flash organization in QSPI mode.

Table 2-2. GD32W51x base address and size for flash memory (QSPI mode)

Block	Name	Address range	size(bytes)
Main flash block (EXT Flash)	-	0x0800 0000 - 0x0A00 0000 ⁽¹⁾ 0x0C00 0000 - 0x0E00 0000 ⁽²⁾	32MB
Bootloader	normal ⁽³⁾	0x0BF4 0000 - 0x0BF4 5FFF	24KB
	secure boot ⁽⁴⁾	0x0FF8 4000 - 0x0FF8 7FFF	16KB
	secure region2 ⁽⁵⁾	0x0FF4 E000 - 0x0FF7 FFFF	200KB
	GSSA ⁽⁴⁾	0x0FF8 0000 - 0x0FF8 3FFF	16KB

Note: (1) TZEN = 0. (2)TZEN= 1. (3)When the TZEN bit is reset, user can choose to boot from this area. (4) When the TZEN bit is set, user can choose to boot from this area. Select secure boot or GSSA please refer to [Boot configuration](#). (5) When the TZEN bit is set or reset, secure region 2 can be accessed by other bootloaders, but booting from this area is not supported. (6) The bootloader block cannot be programmed or erased by user.

2.4.2. Read operations

The flash can be addressed directly as a common memory space.

RTDEC function

RTDEC function means that when reading data from Flash, it can be decrypted in real time according to the EFUSE module's configuration of AES algorithm. (Data written to Flash is encrypted already). There have on-time decrypt function when AESEN bit in EFUSE_USER_CTL register is set. This is implement by hardware on-time and invisible by software. The AES key use AESKEY[31:0] bits in EFUSE_AES_KEY to set. The initial vector is Address[23: 0] / 16.

In FMC mode, fetch data use SIP Flash. Start address is 0x08000000 (non-secure) / 0x0C000000 (secure). Size is 2MB. It can support RTDEC function.

In QSPI mode, fetch data use EXT Flash. Start address is 0x08000000 (non-secure) / 0x0C000000 (secure). Size is 32MB. It can support RTDEC function.

NO- RTDEC function

In FMC mode, for SIP Flash, up to four areas can be configured without RTDEC function by FMC_NODECx (x=0,1,2,3), even if AESEN bit in EFUSE_USER_CTL register is set.

Note: QSPI mode does not support NO-RTDEC function.

Read add offset fuction

In order to meet the needs of Wi-Fi OTA fuction, the read offset function can be configured to increase the bus initial address by add an offset and then read it from FMC. After setting the read offset value and area by configuring FMC_OFRG and FMC_OFVR, reading the value of the source address is equivalent to reading the value of the add offset address. Both SIP Flash memory in FMC mode and EXT Flash memory in QSPI mode support this function when reading form 0x0800 0000 (0x0C00 0000). If user need to configure other functions, the RTDEC area configuration needs to use the source address, and the secure mark area configuration uses the offset address. The setting of the secure mark area has nothing to do with offset. If user need to add secure mark in the offset area, user need to add mark to add offset address instead of the source address. For specific settings of secure mark, please refer to [TrustZone security protection.](#)

Note: (1) Adding offset area does not support configure into NO-RTDEC area. (2) The offset function only supports read operation, not support program operation and erase operation.

2.4.3. Unlock the FMC_CTL/FMC_SECCTL register

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

The OBWEN 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 OBWE N bit in the FMC_CTL register to 1. The software can reset OBWEN bit to 0 to protect the FMC_SECMCFGx (x=0,1,2,3)/FMC_NODECx (x=0,1,2,3)/FMC_OFRG/FMC_OFVR registers .

Note: (1) As long as the wrong key is written to FMC_CTL/FMC_SECCTL, a bus error will be generated, regardless of whether FMC_PRIV bit in FMC_PRIVCFG is set or not. (2) Write wrong key to OBKEY does not generate a bus error.

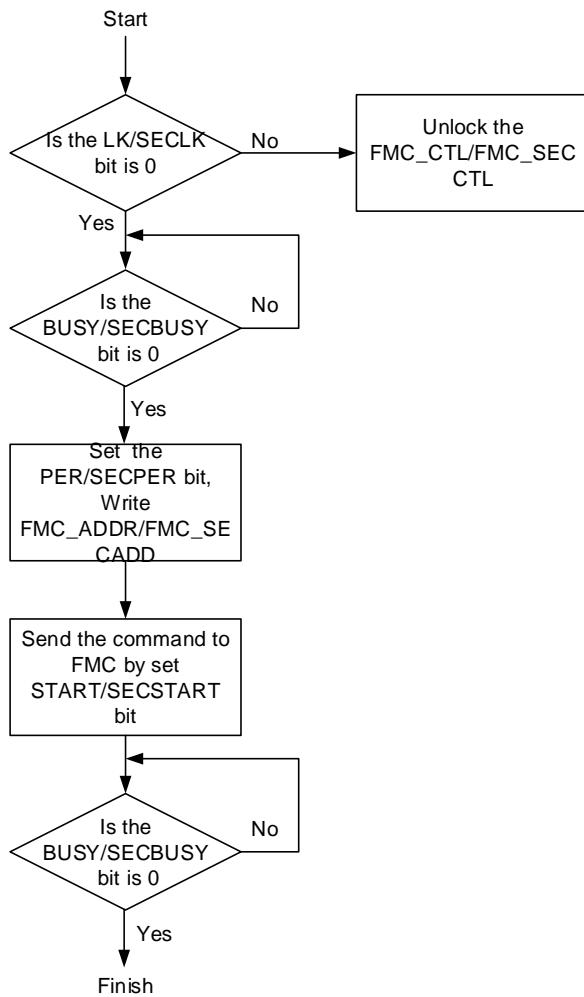
2.4.4. Page erase

The FMC provides a page erase function which is used to initialize the contents of a main Flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. Operate secure page or non-secure page using secure or non-secure registers. The following steps show the access sequence of the registers for a page erase operation.

- Unlock the FMC_CTL/FMC_SECCTL register if necessary.
- Check the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register to confirm that no Flash memory operation is in progress (BUSY/SECBUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the PER/SECPER bit in the FMC_CTL/FMC_SECCTL register.
- Write the page absolute address (0x08XX XXXX/0x0CXX XXXX) into the FMC_A DDR/FMC_SECADDR registers.
- Send the page erase command to the FMC by setting the START/SECSTART bit in the FMC_CTL/FMC_SECCTL register.
- Wait until all the operations have finished by checking the value of the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register.
- Read and verify the page using a DBUS access if required.

When the operation is executed successfully, the ENDF/SECENDF bit in the FMC_STAT/FMC_SECSTAT register is set, and an interrupt will be triggered by FMC if the ENDIE/SECENDIE bit in the FMC_CTL/FMC_SECCTL 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 ENDIE/SECENDIE bit in the FMC_CTL/FMC_SECCTL register is set. The software can check the WPERR/SECWPERR bit in the FMC_STAT/FMC_SECSTAT register to detect this condition in the interrupt handler. The following figure [Figure 2-2 Process of page erase operation](#) shows the page erase operation flow.

Figure 2-2. Process of page erase operation



2.4.5. Mass erase

The FMC provides a complete erase function which is used to initialize the main Flash block contents. Operate secure Flash or non-secure Flash using secure or non-secure registers. This erase can affect entire Flash block by setting the MER/SECMER bit to 1 in the FMC_CTL/FMC_SECCTL register. The following steps show the mass erase register access sequence.

- Unlock the FMC_CTL/FMC_SECCTL register if necessary.
- Check the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register to confirm that no Flash memory operation is in progress (BUSY/SECBUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the MER bit in the FMC_CTL/FMC_SECCTL register if erase entire Flash.
- Send the mass erase command to the FMC by setting the START/SECSTART bit in the FMC_CTL/FMC_SECCTL register.
- Wait until all the operations have been finished by checking the value of the

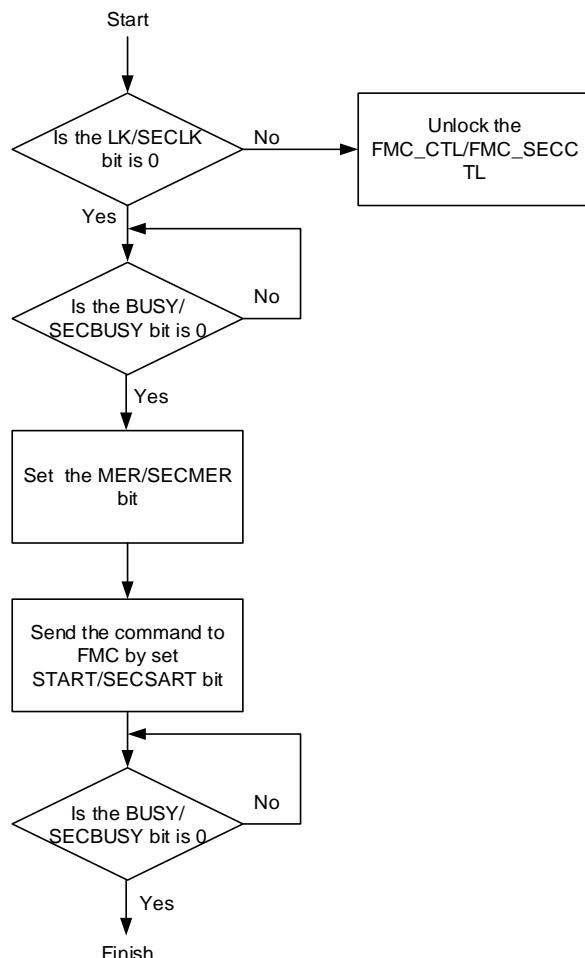
BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register.

- Read and verify the Flash memory using a DBUS access if required.

When the operation is executed successfully, the ENDF/SECENDF bit in the FMC_STAT/FMC_SECSTAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL/FMC_SECCTL 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/SECERRIE bit in the FMC_CTL/FMC_SECCTL register is set. The software can check the WPERR/SECWPERR bit in the FMC_STAT/FMC_SECSTAT register to detect this condition in the interrupt handler.

The following figure [Figure 2-3. Process of mass erase operation](#) indicates the mass erase operation flow.

Figure 2-3. Process of mass erase operation



2.4.6. Main flash programming

The FMC provides a 32-bit word programming function by DBUS which is used to modify the main Flash memory contents. Operate secure Flash or non-secure Flash using secure or non-secure registers.

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

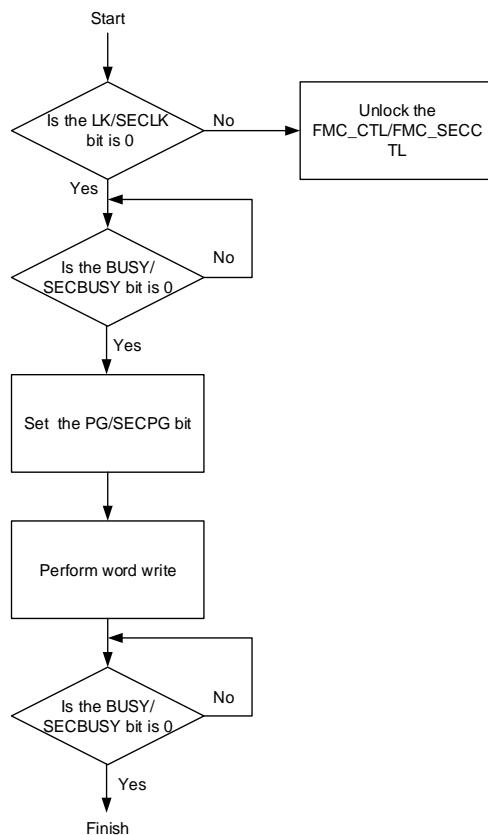
- Unlock the FMC_CTL/FMC_SECCTL register if necessary.
- Check the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register to confirm that no Flash memory operation is in progress (BUSY/SECBUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the PG/SECPG bit in the FMC_CTL/FMC_SECCTL register.
- Write the data to be programmed with desired absolute address (0x08XX XXXX/0x0CXX XXXX).
- Wait until all the operations have been finished by checking the value of the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register.
- Read and verify the Flash memory using a DBUS access if required.

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

Each word can be programmed only one time after erase and before next erase. Note that the PG/SECPG bit must be set before the word programming operation.

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

In these conditions, a Flash operation error interrupt will be triggered by the FMC if the ERRIE/SECERRIE bit in the FMC_CTL/FMC_SECCTL register is set. The software can check WPERR/SECWPERR bit in the FMC_STAT/FMC_SECSTAT register to detect which condition occurred in the interrupt handler. The following figure [Figure 2-4. Process of word program operation](#) shows the word programming operation flow.

Figure 2-4. Process of word program operation


Note: 1. In order to meet Wi-Fi OTA requirements, support continuous program when set PG bit. 2. If user do not do read operation before program, it can realize the wired-AND of the data to be programmed and the data in Flash. 3. In order to achieve the fastest programming speed, user need to follow the following points when programming. (1) 32-bit program and the address is continuous. (2) Program continuously on the bus, there must be no read Flash between two write operation. (3) The interval between two write operations cannot exceed $64 \times T_{hclk}$ (Flash clock is $hclk/2$).

2.4.7. Option bytes

Option bytes description

The option bytes description is shown in the [Table 2-3. Option bytes](#). The option bytes are configured according to the requirements of the application.

Table 2-3. Option bytes

Name	Register map
15: TZEN	Option byte register (FMC_OBR)
12: SRAM1_RST	

Name	Register map
[7:0]: SPC[7:0]	
[15:0]: USER[15:0]	<i>Option byte user register (FMC_OBUSER)</i>
[25:16]: SECMO_EPAGE[9:0] [15:0]: SECMO_SPAGE[9:0]	<i>Secure mark configuration register 0 (FMC_SECM CFG0)</i>
31: DMP0EN [25:16]: DMP0_EPAGE[9:0]	<i>Secure dedicated mark protection register 0 (FMC_DMP0)</i>
[25:16]: WRP0_EPAGE[9:0] [15:0]: WRP0_SPAGE[9:0]	<i>Option byte write protection area register 0 (FMC_OBWRP0)</i>
[25:16]: SEC1_EPAGE[9:0] [15:0]: SEC1_SPAGE[9:0]	<i>Secure mark configuration register 1 (FMC_SECM CFG1)</i>
31: DMP1EN [25:16]: DMP1_EPAGE[9:0]	<i>Secure dedicated mark protection register 1 (FMC_DMP1)</i>
[25:16]: WRP1_EPAGE[9:0] [15:0]: WRP1_SPAGE[9:0]	<i>Option byte write protection area register 1 (FMC_OBWRP1)</i>

Note: If there are option bytes in the chip, operate the registers in the table according to the method of operating the option bytes. If there are no option bytes in the chip, operate the registers in the table according to the method of operating registers.

Option bytes modify

To modify the user options value, follow the procedure below:

- Unlock the FMC_CTL/FMC_SECCTL register if necessary.
- Check the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register to confirm that no flash memory operation is in progress (BUSY/SECBUSY equals to 0)
- 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.
- Write the desired options value in the options registers.
- Set the OBSTART bit in the FMC_CTL register.
- Wait until all the operations have finished by checking the value of the BUSY/SECBUSY bit in the FMC_STAT/FMC_SECSTAT register.
- Set the OBRLD option bit or launch a system reset to start option bytes loading.

Option bytes error

Some of the option bytes field must respect specific rules before being updated with new values.

1. TZEN option bit

Only when the Flash is in no protection state, the TZEN bit can be set to 1 to enable Trustzone. Otherwise, the OBERR bit is set.

The TZEN bit must be cleared at the same time when the Flash security protection r

eturns to unprotected, and Trustzone is disabled. (from protection level 1 to no protection or from protection level 0.5 to no protection). Otherwise, the OBERR bit is set.

2. SECMCFG_x_SPAGE[6:0] (x=0,1), SECMCFG_x_EPAGE[6:0] (x=0,1), DMP_x_EPAGE[6:0] (x=0,1), DMP_xEN (x=0,1) bits

It can only be modified when DMP_x_ACCFG (x=0,1) bit is cleared. When it is set, option bytes are locked and can not be modified until next system reset. If the user try to modify one of the those option bytes while DMP_x_ACCFG (x=0,1) bit is set, the option bytes modification is discarded without error flag.

3. When DMP_xEN (x = 0,1) is set to 1, an invalid secure DMP area is defined (DMP_x_EPAGE>SECM_x_EPAGE), the OBERR is set and the option bytes modification will be discarded.

2.4.8. TrustZone security protection

If there are option bytes, the global TrustZone system security is activated by setting the TZEN bit in FMC_OBR register. If there are no option bytes, the global TrustZone system security is activated by setting the TZEN bit in EFUSE_TZCTL register.

When the TrustZone is active (TZEN=1), additional security features are available:

- 4 secure mark registers define SIP Flash secure areas.
- DMP register defines secure dedicated mark protection areas.
- An additional security protection level: protection level 0.5.
- Erase or program operation can be performed in secure or non-secure mode with associated configuration bit.
- The secure mark of EXT Flash access by QSPI interface is implemented in the TZSPC module.

When the TrustZone is disabled (TZEN=0), the above features are deactivated and all secure registers are RAZ/WI (read as zero/write ignore).

Secure mark area

When TrustZone security is active (TZEN=1), apart of the SIP Flash can be protected against non-secure read and write access by configure FMC_SECMCFG_x(x=0, 1, 2, 3).

Secure mark area 0/1: If there are option bytes, defined by option bytes. Or else defined by secure software on-the-fly.

Secure mark area 2/3: Defined by secure software on-the-fly.

Secure dedicated mark protection area (DMP)

The secure DMP area is part of a secure area based on Flash secure mark. Configure the DMP_x_ACCFG (x=0,1) bit in FMC_DMPCTL register to set access permission to the DMP

area. When DMPx_ACCFG ($x=0,1$) bit is set, the code in the DMP area can only be executed once and does not support read and write operations of data and read operation of instructions until the next system reset. The DMPx_ACCFG ($x=0,1$) bit can only be cleared by a system reset.

The secure DMP area size is defined by the end page offset using the DMPx_EPAGE in FMC_DMPx ($x=0,1$) while the start page offset is already defined by SECMx_SPAGE in FMC_SECMCFGx ($x=0,1$) registers. If DMPx_ACCFG bit is set, DMPxEN and DMPx_EPAGE can not be modified until the next system reset. If DMPx_EPAGE > SECMx_EPAGE, an invalid secure DMP area is defined. The SECERR flag bit will be set and the FMC_DMPx ($x=0,1$) register modifications will not be valid at the time.

DMP area 0/1: If there are option bytes, defined by option bytes. Or else defined by secure software on-the-fly.

Note: This function is only available in FMC mode.

Flash security attribute state

Flash is secure when at least one security area is defined by FMC_SECMCFGx($x=0,1,2,3$) registers.

2.4.9. 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.

Note: FMC mode and QSPI mode implements the same security protection strategy.

No protection

If there are option bytes, when SPC[7:0] bits in FMC_OBR is set to 0xAA, after the system is reset, the Flash memory will be in no protection state. If there are no option bytes, configure FP[7: 0] bits in EFUSE_FP_CTL to level 0, after the system is reset, the Flash memory will be in no protection state. The main Flash are accessible by all operations. The SRAM1 and the backup registers are also accessible by all operations. In GSSA mode, the debug access is disabled.

Protection level 0.5 (available only when TZEN = 1)

If there are option bytes, when SPC[7:0] bits in FMC_OBR is set to 0x55, after the system is reset, the Flash memory will be in protection level 0.5 state. If there are no option bytes, configure FP [7: 0] in EFUSE_FP_CTL to level 0.5, after the system is reset, the Flash memory will be in protection level 0.5 state. All read and write operations (if no write protection is set) from/to the non-secure Flash memory are possible. The debug access to secure area is prohibited. Debug access to non-secure area remains possible. In GSSA mode, the debug access is disabled.

Protection level 1

If there are option bytes, when TZEN = 0 and SPC[7:0] bits in FMC_OBR is set to to any value except 0xAA, after the system is reset, the Flash memory will be in protection level 1 state. When TZEN = 1 and the SPC byte is set to to any value except 0xAA and 0x55, after the system is reset, the Flash memory will be in protection level 1 state. If there are no option bytes, configure FP [7: 0] in EFUSE_FP_CTL to level 1, after the system is reset, protection level 1 performed.

Whether Trustzone is enabled or not, Flash, SRAM and backup registers are accessible to the code booted from the Flash, including read/program/erase all operations.

When TrustZone is enabled (TZEN=1):

When the CPU is in a non-secure state, non-secure debugging can be performed. When an intrusion is detected, the Flash, backup registers, and SRAM1 are completely inaccessible. Read and write access to Flash or backup registers or SRAM1 or RTDEC area can result in bus errors and Hard Fault interrupt. When the security protection level is set to level 1 and debug access is detected, an intrusion will be detected.

When booting from GSSA, the debug access is disabled.

When TrustZone is enabled, it is no longer be able to boot from SRAM.

When TrustZone is disabled (TZEN=0):

Flash, backup registers, and SRAM1 are completely inaccessible in debug mode or whe code booting from RAM or from bootloader. In these startup modes, intrusions are detected and read and write access to Flash or backup SRAM results in bus error and Hard Fault interrupt.

Table 2-4. Flash secure operation under different protection levels when TrustZone is active (TZEN=1)

Access type		Fetch	Read	Write	Page erase
SPC0, SPC 0.5, SPC 1 ⁽¹⁾	Non-secure page	Bus error	all read data is 0, cause Flash illegal access event	write invalid, SECWPERR flag set, Flash illegal access event	
	DMP area (DMPxEN=1 and DMPx_ACCDIS=1)	all read data is 0		write invalid, SECWPERR flag set	
	Secure page	OK		no write protection pages: OK write protection pages: WI and SECWPERR flag set	

Access type		Fetch	Read	Write	Page erase
SPC 1 ⁽²⁾	Non-secure page or secure page		Bus error	write invalid, SECWPERR flag set	

Table 2-5. Flash non-secure operation under different protection levels when TrustZone is active (TZEN=1)

Access type		Fetch	Read	Write	Page erase
SPC0, SPC 0.5, SPC 1 ⁽¹⁾	Non-secure page		OK	no write protection pages: WI and WPERR flag set	
	DMP area (DMPxEN=1 and DMPx_ACCDIS=1)	Bus error	all read data is 0, Flash illegal access event	write invalid, WPERR flag set, Flash illegal access event	
	Secure page				
SPC 1 ⁽²⁾	Non-secure page or secure page		Bus error	write invalid, SECWPERR flag set	

Note: (1)Booting from Flash and no debug access. (2) Debug access is detected.

Table 2-6. Flash operation under different protection levels when TrustZone is disable (TZEN=0)

Access type		Fetch	Read	Write	Page erase
SPC0, SPC 0.5, SPC 1 ⁽¹⁾			OK	no write protection pages: OK write protection pages: WI and WPERR flag set	
SPC 1 ⁽²⁾			Bus error	write invalid, WPERR flag set	

Note: (1)Booting from Flash and no debug access. (2) Debug access is detected.

Table 2-7. Flash mass erase operation under different protection levels when TrustZone is active (TZEN=1)

Access type		SPC 0.5, SPC 1 ⁽¹⁾				SPC 1 ⁽²⁾	
		Non-secure page	Secure Flash		Mix non-secure page and Secure page		
			DMP area (DMPxEN=1 and DMPx_ACCFG=1)	Others ⁽³⁾			
secure	Mass erase	w rite invalid, SECWPERR flag set, Flash illegal access event	w rite invalid and SECWPERR flag set	no w rite protection pages: OK w rite protection pages: WI and SECWPERR flag set	w rite invalid, SECWPERR flag set, Flash illegal access event	w rite invalid, SECWPERR flag set	
Non-secure	Mass erase	no w rite protection pages: OK w rite protection pages: WI and WPERR flag set	all read data is 0, Flash illegal access event w rite invalid, WPERR flag set, Flash illegal access event			w rite invalid, WPERR flag set	

Note: (1)Booting from Flash and no debug access. (2) Debug access is detected. (3) Others refer to other Flash security configurations that are different from the Flash security configuration described for DMP protection.

In QSPI mode, when the user accesses the external Flash memory in violation of the Flash security attribute configured by secure mark, if it is a fetch instruction, an error response is returned, and data 0 is returned. If it is not a fetch instruction, data 0 is returned.

Modify security protection level when Trust zone is enabled

It is easy to move from security protection level 0 or level 0.5 to level 1 by changing the value of the SPC[7:0] in FMC_OBR to any value except for 0x55 or 0xAA.

When the SPC[7:0] is programmed to the value 0xAA to move from level 0.5 or from level 1 to level 0, a mass erase of the Flash main memory is performed. The backup registers and all SRAMs are also erased.

When the SPC[7:0] is programmed to the value 0x55 to move from level 1 to level 0.5, a partial mass erase of Flash main memory is performed. Only non-secure mark areas are erased. The backup registers and all SRAMs are also erased.

Note: Only when there are option bytes, security protection level can be modified.

2.4.10. Write protection

The FMC provides a write protection function to prevent program/erase function 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/SECWPERR bit in the FMC_STAT/FMC_SECSTAT register will be set by the FMC. If the WPERR/SECWPERR bit is set and the ERRIE/SECERRIE 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. When write protection is set in EFUSE, first 32KB are write protect, then a system reset is necessary. When there are option bytes, the write-protected area defined in the option bytes (FMC_OBR) is also write-protected.

Note: This function is only available in FMC mode.

2.4.11. Flash privileged and unprivileged mode

The Flash registers can be read and written by privileged and unprivileged accesses depending on PRIV bit in FMC_PRIVCFG register. When the FMC_PRIV bit is reset, Flash registers can be read and written by both privileged and unprivileged access. When the FMC_PRIV bit is set, all Flash registers could be read and written by privileged access only. Unprivileged access to a privileged registers is RAZ/WI.

2.4.12. FLASH interrupts

There includes secure interrupts and non-secure interrupts.

Secure interrupts: secure end of operation/secure operation error.

Non-secure interrupts: non-secure end of operation/non-secure operation error.

Table 2-8. Flash interrupt requests (non-secure)

Flag	Description	Clear method	Interrupt enable bit
ENDF	end of operation	Write 1 to corresponding bit in FMC_STAT register	ENDIE
WPERR	erase/program on protected pages		ERRIE

Table 2-9. Flash interrupt requests (secure)

Flag	Description	Clear method	Interrupt enable bit
SECENDF	end of operation	Write 1 to corresponding bit in FMC_SECSTAT register	SECENDIE
SECWPERR	erase/program on protected pages		
SECERR	an invalid secure DMP area is defined (DMPx_PEND>SEC Mx_PEND)		SECERRIE

2.5. Register definition

FMC secure base address: 0x5002 2000
 FMC non-secure base address: 0x4002 2000

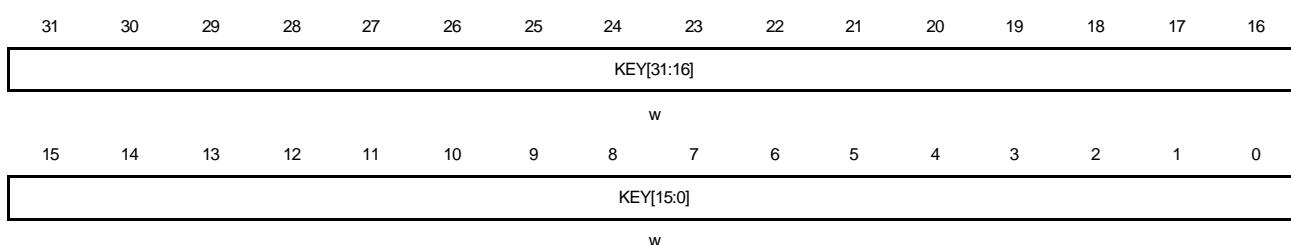
2.5.1. Unlock key register (FMC_KEY)

Address offset: 0x04

Reset value: 0x0000 0000

This register is non-secure. Protected against non-privileged access when FMC_PRIV = 1.

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



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

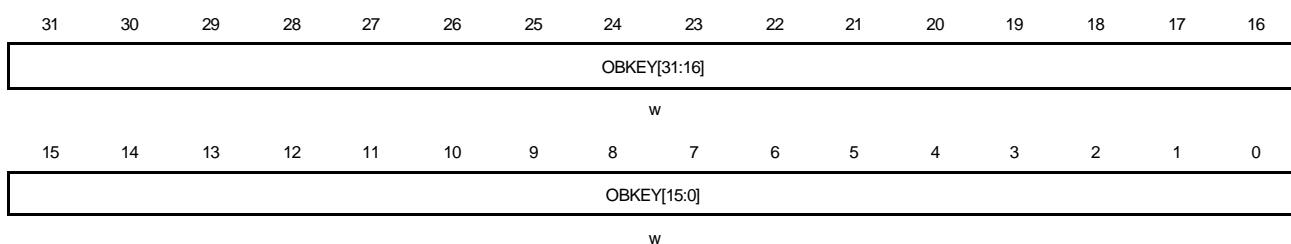
2.5.2. Option byte unlock key register (FMC_OBKEY)

Address offset: 0x08

Reset value: 0x0000 0000

This register is non-secure. Protected against non-privileged access when FMC_PRIV =1.

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



Bits	Fields	Descriptions
31:0	OBKEY[31:0]	<p>These bits are only be written by software.</p> <p>Write OBKEY [31:0] with keys to unlock OBWEN bit in FMC_CTL register.</p>

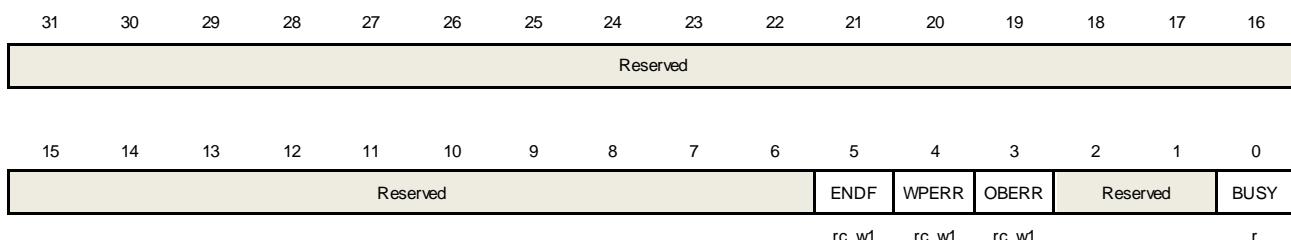
2.5.3. Status register (FMC_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register is non-secure. Protected against non-privileged access when FMC_PRIV=1.

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
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	OBERR	Option bytes error flag bit (If there are option bytes) If user set the TZEN bit when the Flash is not in no protection state, this bit is set. If user does not clear the TZEN bit at the same time when the Flash security protection returns to unprotected, this bit is set. If an invalid secure DMP area is defined (DMPx_EPAGE > SECm_EPAGE), this bit is set and the FMC_DMPx modification is discarded. The software can clear it by writing 1.
2:1	Reserved	Must be kept at reset value.
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.

2.5.4. Control register (FMC_CTL)

Address offset: 0x10

Reset value: 0x0000 0080

This register is non-secure. Protected against non-privileged access when FMC_PRIV=1.

This register can only be written when the BUSY bit in FMC_CTL and SECBUSY bit in

FMC_SECCTL are reset. Otherwise, the write access is stalled till BUSY bit in FMC_CTL and SECBUSY bit in FMC_SECCTL are 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
OBRD	OBSTART	Reserved	ENDIE	Reserved	ERRIE	OBWEN	Reserved	LK	START	Reserved	MER	PER	PG		

rw rw rw rw rw rs rs rw rw rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	OBRD	Option byte reload bit This bit is set by softw are 0: no effect 1: force option byte reload.
14	OBSTART	Option bytes modification start bit 0: no effect 1: trigger an option bytes operation. This bit can only be written if OBWEN bit is set. This bit is only set by software, and is cleared when the BUSY bit is cleared in FMC_STAT.
13	Reserved	Must be kept at reset value.
12	ENDIE	End of operation interrupt enable bit This bit is set or cleared by softw are 0: no interrupt generated by hardw are. 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 softw are 0: no interrupt generated by hardw are. 1: error interrupt enable
9	OBWEN	FMC_OBR / FMC_OBUSER / FMC_SECMx (x=0,1,2,3) / FMC_NODECx (x=0,1,2,3) / FMC_OFRG / FMC_OFVR / FMC_OBWRPx (x=0,1) write enable bit This bit is set by hardw are when right sequence written to FMC_OBKEY register. This bit can be cleared by softw are.
8	Reserved	Must be kept at reset value.
7	LK	FMC_CTL lock bit This bit is cleared by hardw are when right sequence written to the FMC_KEY

register. This bit can be set by software.

6	START	Send erase command to FMC This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5:3	Reserved	Must be kept at reset value.
2	MER	Main Flash mass erase command bit This bit is set or cleared by software 0: no effect 1: main Flash mass erase command
1	PER	Main Flash page erase command bit This bit is set or clear by software 0: no effect 1: main Flash page erase command
0	PG	Main Flash program command bit This bit is set or clear by software 0: no effect 1: main Flash program command

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

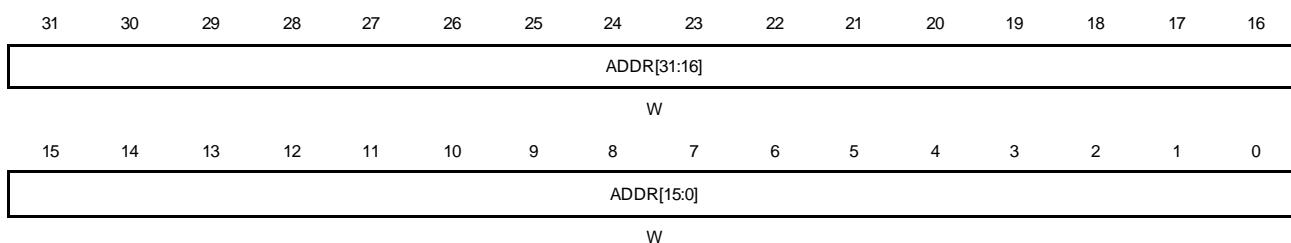
2.5.5. Address register (FMC_ADDR)

Address offset: 0x14

Reset value: 0x0000 0000

This register is non-secure. Protected against non-privileged access when FMC_PRIV=1.

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.

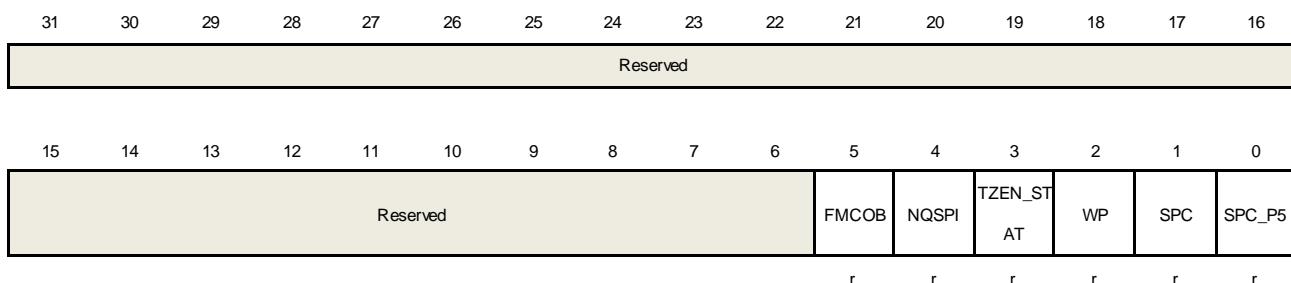
2.5.6. Option byte status register (FMC_OBSTAT)

Address offset: 0x1C

Reset value: 0x0XXX XXXX

This register is non-secure. Protected against non-privileged access when FMC_PRIV = 1.

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	FMCOB	Whether the option byte exist or not. 0: There are no option bytes. 1: There are option bytes.
4	NQSPI	Memory structure is FMC mode or QSPI mode 0: QSPI mode: EXT Flash. 1: FMC mode: SIP Flash.
3	TZEN_STAT	Trust zone state 0: Trust zone is disabled. 1: Trust zone is enabled.
2	WP	Write/erase protection state 0: write/erase protection is reset 1: write/erase protection is set
1	SPC	Security protection level 1 state 0: protection level 1 is reset 1: protection level 1 is set
0	SPC_P5	Security protection level 0.5 state 0: protection level 0.5 is reset 1: protection level 0.5 is set

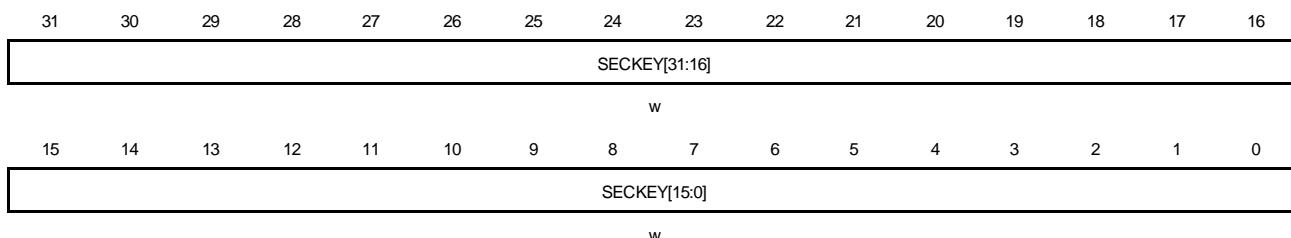
2.5.7. Secure Unlock key register (FMC_SECKEY)

Address offset: 0x24

Reset value: 0x0000 0000

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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



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

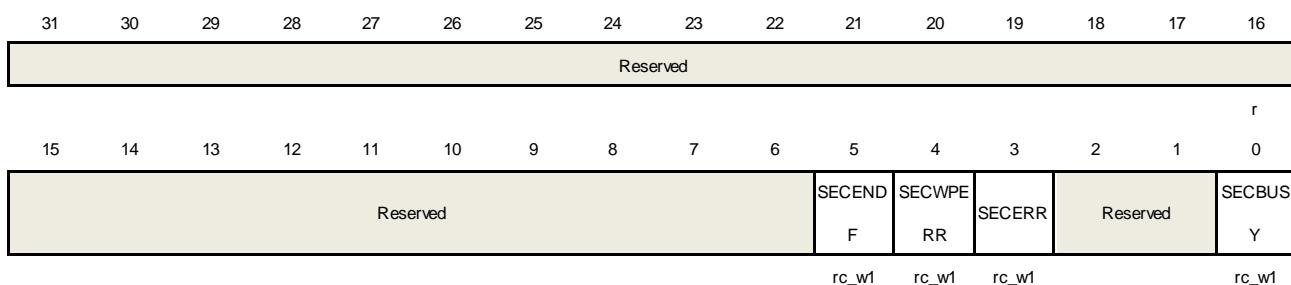
2.5.8. Secure status register (FMC_SECSTAT)

Address offset: 0x2C

Reset value: 0x0000 0000

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	SECENDF	End of operation flag When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	SECWPERR	Erase/Program protection error flag When erase/program on protected pages, this bit is set by hardware.

The software can clear it by writing 1.

3	SECERR	Secure error flag (If there are no option bytes) If an invalid secure DMP area is defined (DMPx_EPAGE > SECm_EPAGE), this bit is set and the FMC_DMPx (x=0,1) modification is discarded. The software can clear it by writing 1.
2:1	Reserved	Must be kept at reset value.
0	SECBUSY	The Flash is busy 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.

2.5.9. Secure Control register (FMC_SECCTL)

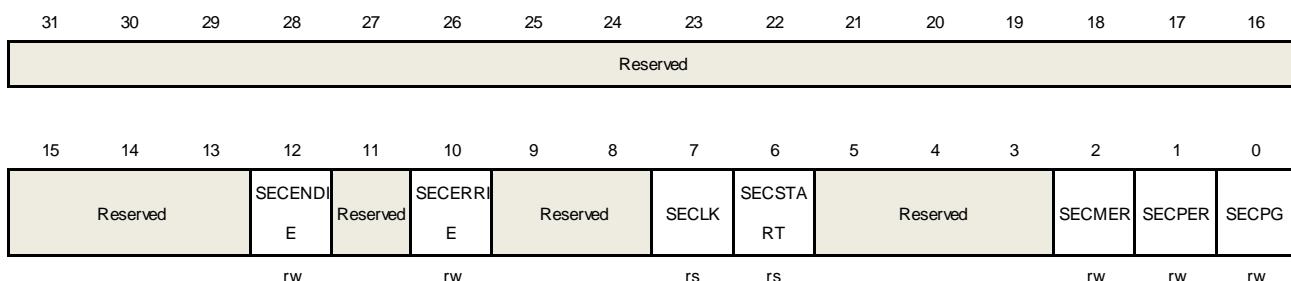
Address offset: 0x30

Reset value: 0x0000 0080

This register can only be written when the SECBSY and NSBSY are reset. Otherwise, the write access stalls till SECBSY and NSBSY are reset.

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	SECENDIE	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	SECERRIE	Error interrupt enable bit This bit is set or cleared by software. 0: no interrupt generated by hardware

		1: error interrupt enable
9:8	Reserved	Must be kept at reset value.
7	SECLK	<p>FMC_SECCTL lock bit</p> <p>This bit is cleared by hardware when right sequence written to FMC_SECKEY register.</p> <p>This bit can be set by software.</p>
6	SECSTART	<p>Send erase command to FMC bit</p> <p>This bit is set by software to send erase command to FMC.</p> <p>This bit is cleared by hardware when the BUSY bit is cleared.</p>
5:3	Reserved	Must be kept at reset value.
2	SECMER	<p>Main Flash mass erase command bit</p> <p>This bit is set or cleared by software.</p> <p>0: no effect</p> <p>1: main Flash mass erase command</p>
1	SECPER	<p>Main Flash page erase command bit</p> <p>This bit is set or clear by software.</p> <p>0: no effect</p> <p>1: main Flash page erase command</p>
0	SECPG	<p>Main Flash program command bit</p> <p>This bit is set or clear by software.</p> <p>0: no effect</p> <p>1: main Flash program command</p>

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

2.5.10. Secure Address register (FMC_SECADDR)

Address offset: 0x34

Reset value: 0x0000 0000.

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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
SECADDR[31:16]															
W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECADDR[15:0]															
W															

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

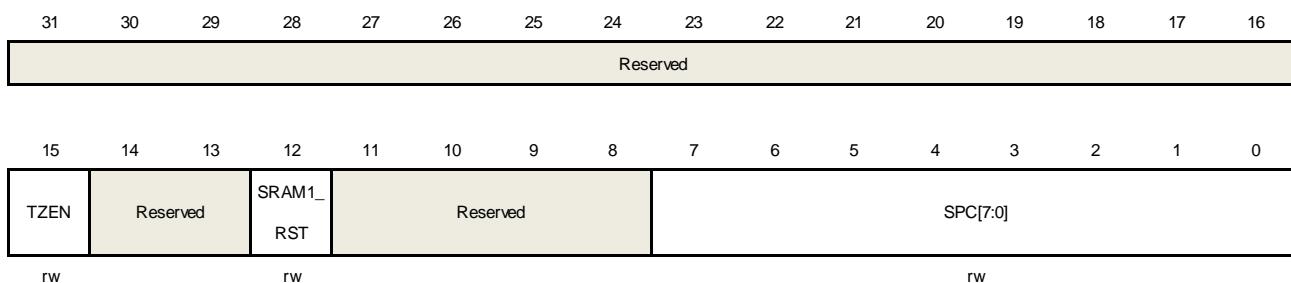
2.5.11. Option byte register (FMC_OBR)

Address offset: 0x40

Reset value: 0xFFFF XXXX (Register bits 0 to 31 are loaded with values from Flash memory when OBRLD is set or system reset. The loading condition of the SRAM1_RST bit must be power-on reset.)

This register can only be written if OBWEN bit is set. This register is non-secure. It can be read and written by both secure and non-secure access. This register can be protected against non-privileged access when PRIV=1 in the FMC_PRIV register.

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	TZEN	Trust zone enable bit 0: Disable Trust zone function. Effective after system reset. 1: Enable Trust zone function. Effective after system reset. Note: If there are option bytes, this bit decides whether to enable Trust zone or not , otherwise TZEN bit in EFUSE_TZ_CTL register decides.
14:13	Reserved	Must be kept at reset value.
12	SRAM1_RST	SRAM1 reset enable bit 0: no effect 1: clear SRAM1 data automatically. Effective after system reset.
11:8	Reserved	Must be kept at reset value.
7:0	SPC[7:0]	Flash security protection value. Effective after system reset. Note: If there are option bytes, the security protection of the Flash memory is subject to this bits field, otherwise it is subject to the FP[7:0] bits in the

EFUSE_FP_CTL register.

2.5.12. Option byte user register (FMC_OBUSER)

Address offset: 0x44

Reset value: 0xFFFF XXXX (Register bits 0 to 31 are loaded with values from Flash memory when OBRLD is set or system reset.)

This register can only be written if OBWEN bit is set. This register is non-secure. It can be read and written by both secure and non-secure access. This register can be protected against non-privileged access when PRIV=1 in the FMC_PRIV register.

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
USER[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	USER[15:0]	Option byte USER value. Note: If there are option bytes, USER is subject to this bit field configuration, otherwise it is subject to the configuration in the EFUSE_USER_CTL register.

2.5.13. Secure mark configuration register 0 (FMC_SECMCFG0)

Address offset: 0x48

Reset value: 0x03FF 0000/ 0xFFFF XXXX (When there are option bytes, register bits 0 to 31 are loaded with values from Flash memory when OBRLD is set or system reset. When there no option bytes, the reset value is kept at 0x03FF 0000.)

This register can only be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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								SECM0_EPAGE[9:0]							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SECM0_SPAGE[9:0]							

rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	SECM0_EPAGE[9:0]	End page of secure mark area 0. If DMP0_ACCFG is 1, this bits cannot be modified. If SECM0_EPAGE < DMP0_EPAGE, SECERR bit will be set and discard this write.
15:10	Reserved	Must be kept at reset value.
9:0	SECM0_SPAGE[9:0]	Start page of secure mark area 0. If DMP0_ACCFG is 1, this bits cannot be modified.

2.5.14. Secure dedicated mark protection register 0 (FMC_DMP0)

Address offset: 0x4C

Reset value: 0x0000 0000

This register can only be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when PRIV=1.

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
DMP0EN		Reserved										DMP0_EPAGE[9:0]			
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31	DMP0EN	DMP area 0 enable 0: disable 1: enable
30:26	Reserved	Must be kept at reset value.
25:16	DMP0_EPAGE[9:0]	End page of DMP area 0.
15:0	Reserved	Must be kept at reset value.

2.5.15. Option byte write protection area register 0 (FMC_OBWRP0)

Address offset: 0x50

Reset value: 0xFFFF XXXX (Register bits 0 to 31 are loaded with values from Flash memory

when OBRLD is set or system reset.)

This register can only be written if OBWEN bit is set. This register is non-secure. It can be read and written by both secure and non-secure access. This register can be protected against non-privileged access when FMC_PRIV=1.

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								WRP0_EPAGE[9:0]							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WRP0_SPAGE[9:0]							
								rw							

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	WRP0_EPAGE[9:0]	End page of write protection area 0
15:10	Reserved	Must be kept at reset value.
9:0	WRP0_SPAGE[9:0]	Start page of write protection area 0

2.5.16. Secure mark configuration register 1 (FMC_SECMCFG1)

Address offset: 0x54

Reset value: 0x0000 03FF / 0xXXXX XXXX (When there are option bytes, register bits 0 to 31 are loaded with values from Flash memory when OBRLD is set or system reset. When there no option bytes, the reset value is kept at 0x0000 03FF.)

This register can not be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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								SECM1_EPAGE[9:0]							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SECM1_SPAGE[9:0]							
								rw							

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.

25:16	SECM1_EPAGE[9:0]	End page of secure mark area 1. If DMP1_ACCFG is 1, this bits cannot be modified. If SECM1_EPAGE < DMP1_EPAGE, SECERR bit will be set and discard this write.
15:10	Reserved	Must be kept at reset value.
9:0	SECM1_SPAGE[9:0]	Start page of secure mark area 1. If DMP1_ACCFG is 1, this bits cannot be modified.

2.5.17. Secure dedicated mark protection register 1 (FMC_DMP1)

Address offset: 0x58

Reset value: 0x0000 0000

This register can only be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV =1.

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
DMP1EN	Reserved										DMP1_EPAGE[9:0]				
rw															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31	DMP1 EN	DMP area 1 enable 0: disable 1: enable
30:26	Reserved	Must be kept at reset value.
25:16	DMP1_EPAGE[9:0]	End page of DMP area 1.
15:0	Reserved	Must be kept at reset value.

2.5.18. Option byte write protection area register 1 (FMC_OBWRP1)

Address offset: 0x5C

Reset value: 0xFFFF XXXX (Register bits 0 to 31 are loaded with values from Flash memory when OBRD is set or system reset.)

This register can only be written if OBWEN bit is set. This register is non-secure. It can be read and written by both secure and non-secure access. This register can be protected against non-privileged access when FMC_PRIV = 1.

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								WRP1_EPAGE[9:0]							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WRP1_SPAGE[9:0]							
rw															

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	WRP1_EPAGE[9:0]	End page of write protection area 1
15:10	Reserved	Must be kept at reset value.
9:0	WRP1_SPAGE[9:0]	Start page of write protection area 1

2.5.19. Secure mark configuration register 2(FMC_SECMCFG2)

Address offset: 0x60

Reset value: 0x0000 03FF.

This register can not be written if OBWEN bit is set.

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when PRIV=1.

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								SECM2_EPAGE[9:0]							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SECM2_SPAGE[9:0]							
rw															

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	SECM2_EPAGE[9:0]	End page of secure mark area 2.
15:10	Reserved	Must be kept at reset value.
9:0	SECM2_SPAGE[9:0]	Start page of secure mark area 2.

2.5.20. Secure mark configuration register 3 (FMC_SECMCFG3)

Address offset: 0x64

Reset value: 0x0000 03FF.

This register can not be written if OBWEN bit is set.

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when PRIV=1.

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								SECM3_EPAGE[9:0]							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SECM3_SPAGE[9:0]							
rw															

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	SECM3_EPAGE[9:0]	End page of secure mark area 3.
15:10	Reserved	Must be kept at reset value.
9:0	SECM3_SPAGE[9:0]	Start page of secure mark area 3.

2.5.21. NO-RTDEC region register x (FMC_NODECx, x=0,1,2,3)

Address offset: 0x70 + 0x4 * x (x = 0 to 3)

Reset value: 0x0000 03FF

This register can not be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

It is only defined in the secure area and provided for use in the non-secure area. Non-secure can not be defined.

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	NODECx_EPAGE[9:0]	End page of NODEC region x (x=0,1,2,3).
15:10	Reserved	Must be kept at reset value.
9:0	NODECx_SPAGE[9:0]	Start page of NODEC region x (x=0,1,2,3).

2.5.22. Offset region register (FMC_OFRG)

Address offset: 0x80

Reset value: 0x0000 1FFF

This register can not be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:16	OF_EPAGE[12:0]	End page of offset region.
15:13	Reserved	Must be kept at reset value.
12:0	OF_SPAGE[12:0]	Start page of offset region

2.5.23. Offset value register (FMC_OFVR)

Address offset: 0x84

Reset value: 0x0000 0000

This register can not be written if OBWEN bit is set.

This register is secure when TZEN = 1. It can be read and written only by secure access or TZEN = 0. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when PRIV=1.

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				OF_VALUE[12:0]											

rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:0	OF_VALUE[12:0]	Offset value

2.5.24. Secure dedicated mark protection control register (FMC_DMPCTL)

Address offset: 0x8C

Reset value: 0x0000 0000

This register can only be written if OBWEN bit is set.

This register is secure. It can be read and written only by secure access. A non-secure read/write access is RAZ/WI. This register can be protected against non-privileged access when FMC_PRIV = 1.

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												DMP1_A	DMP0_A	CCFG	CCFG

rw rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMP1_ACCFG	DMP area 1 access configuration bit When set, this bit is only cleared by a system reset 0: Access to DMP area 1 is granted 1: Access to DMP area 1 is denied and DMPxEN/DMPx_PEND (x=0,1) modification are denied.
0	DMP0_ACCFG	DMP area 0 access configuration bit When set, this bit is only cleared by a system reset 0: Access to DMP area 0 is granted 1: Access to DMP area 0 is denied and DMPxEN/DMPx_PEND (x=0,1)

modification are denied.

2.5.25. Privilege configuration register (FMC_PRIVCFG)

Address offset: 0x90

Reset value: 0x0000 0000.

This register can be read by both privileged and unprivileged access.

When the system is secure (TZEN = 1), this register can be read by secure and non-secure access. It is write-protected against non-secure write access when the Flash is secure. A non-secure write access is ignored and generates an illegal access event.

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															FMC_PRIV

rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	FMC_PRIV	<p>This bit can be read by both privileged or unprivileged, secure and non-secure access. When set, it can only be cleared by a privileged access.</p> <p>0: All Flash registers can be read and written by privileged or unprivileged access.</p> <p>1: All Flash registers can be read and written by privileged access only.</p> <p>If the Flash is not secure (non-secure area defined), the FMC_PRIV bit can be written by a secure or non-secure privileged access. If the Flash is secure, the FMC_PRIV bit can be written only by a secure privileged access:</p> <ul style="list-style-type: none"> – A non-secure write access is ignored and generates an illegal access event. – A secure unprivileged write access on PRIV bit is ignored.

2.5.26. Product ID register (FMC_PID)

Address offset: 0x100

Reset value: 0xXXXX 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
PID[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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

3. Electronic fuse (EFUSE)

3.1. Overview

The EFUSE controller has efuse macro that store system parameters. As a non-volatile unit of storage, the bit of efuse macro cannot be restored to 0 once it is programmed to 1. According to the software operation, the EFUSE controller can program all bits in the system parameters.

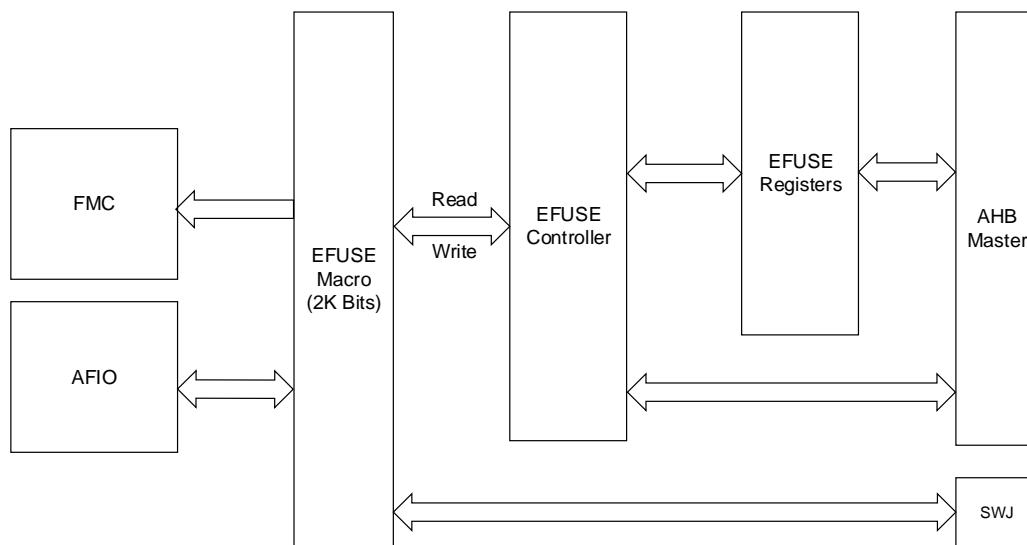
3.2. Characteristics

- One-time programmable nonvolatile EFUSE storage cells organized as 256*8 bit.
- All bits in the efuse cannot be rollback from 1 to 0.
- Can only be accessed through corresponding register.

3.3. Block diagram

Efuse_ctl implements the efuse macro read-program control logic. Efuse_ctl contains efuse_2Kb module that instantiate EFUSE macro.

Figure 3-1. Block diagram of efuse controller



3.4. Function overview

3.4.1. EFUSE architecture

The EFUSE consists of up to 2048 bits storage cells organized into 256 bytes. EFUSE uses 8-bit address encoding. The following table [Table 3-1. EFUSE address mapping](#) shows the address.

Table 3-1. EFUSE address mapping

ADDR [7:0]	EFUSE byte
0000_0000	EFUSE[0]
0000_0001	EFUSE[1]
0000_0010	EFUSE[2]
.	.
.	.
1111_1111	EFUSE[255]

3.4.2. EFUSE macro description

The EFUSE macro stores 18 system parameters, every system parameter has different width.

The following table [Table 3-2. system parameters](#) shows the details of each EFUSE byte.

Table 3-2. system parameters

Parameter	Width /B	Start address	Program-protected	Read-protected	Description	Note
Efuse control	1B	8'd0	Can write multiple times, but can not rollback	Read out after system reset and keep unchanged, bus readable	Control bytes of relevant parameters required for MCU startup. For more details, refer to <u>Control register (EFUSE CTL)</u>	User defined
Trustzone control	1B	8'd1	Can write multiple times, but can not rollback	Read out after system reset and keep unchanged, bus readable	Control bytes of relevant parameters required for Trustzone enable. For more details, refer to <u>Trustzone control register(EFUSE TZCTL)</u>	User defined
Flash protection	1B	8'd2	Can write multiple times, but can not rollback	Read out after system reset and keep unchanged, bus readable	The option of flash protection. For more details, refer to <u>Flash protection control register (EFUSE FP CTL)</u>	User defined
User control	1B	8'd3	Can write multiple times, but can not rollback	Read out after system reset and keep unchanged, bus readable	The option of user control function For more details, refer to <u>User control register (EFUSE USER CTL)</u>	User defined
MCU initialization parameters	12B	8'd4	once-time	Read only once after system reset, bus readable	The relevant parameters required for MCU startup For more details, refer to <u>MCU initialization data register (EFUSE MCU INIT DATA)</u>	User defined
AES key	16B	8'd16	once-time	Read out after system reset and keep unchanged, bus readable	The AES key used to encrypt the firmware image For more details, refer to <u>Firmware AES key register (EFUSE AES KEY)</u>	User defined
RoTPK (or its hash)	32B	8'd32	Write once after system reset	Read out by ROM after system reset, bus readable	Root of Trust Public Key (ECC's public key/the HASH result of the RSA's Public Key) For more details, refer to <u>RoTPK key register (EFUSE ROTPK KEY)</u>	User defined
Debug	8B	8'd64	once-time	Read out by	Initial Attestation Key, which	User

Parameter	Width /B	Start address	Program-protected	Read-protected	Description	Note
password				ROM after system reset, bus readable	used for initial attestation service For more details, refer to <u>Debug password register (EFUSE DP)</u>	define d
IAK/GSSA	64B	8'd72	Write once after system reset	Read out by ROM after system reset, bus readable	Initial Attestation Key, which used for initial attestation service(When the efuse attribute is IAK, it is written by the user, otherwise it is solidified by the manufacturer). For more details, refer to <u>IAK key or GSSA register (EFUSE IAK_GSSA)</u>	User define d / CP
MCU UID	16B	8'd136	can not modify	Read out after system reset and keep unchanged, bus readable	the Unique ID of MCU For more details, refer to <u>Product UID register (EFUSE PUID)</u>	CP
HUK	16B	8'd152	can not modify	Read out after system reset and keep unchanged, bus readable when rom_rd_en valid	Hardware Unique Key, which provides the RoT (Root of Trust) for confidentiality. For more details, refer to <u>HUK key register (EFUSE HUK KEY)</u>	CP
RF data	CCK Power Index	6Bx2	8'd168	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	CCK Tx power control index,different channel have different value
			8'd 174			
	OFDM Power Index	6Bx2	8'd 180	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	OFDM Tx power control index, different channel have different value
			8'd 186			
	Channel Plan	1Bx2	8'd 192	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	Channel plan: WLAN use different channel planning and power settings in different countries and regions.
	Crystal	1Bx2	8'd 154			
				Read out after	Crystal calibration value	Customer trim

Parameter	Width /B	Start address	Program-protected	Read-protected	Description	Note
Calibration		8'd 155	after system reset	system reset, bus readable when rom_rd_en valid		mer trim
Thermal Meter	1Bx2	8'd 196	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	Thermal meter value	Customer trim
		8'd 197				
IQK/LCK	1Bx2	8'd 198	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	RF IQ cal/LC cal startup mechanism	Customer trim
		8'd 199				
MAC Address	6Bx2	8'd 200	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	WLAN MAC address	Customer trim
		8'd 206				
IC RF type	1Bx2	8'd 212	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid	IC RF type related options	Customer trim
		8'd 213				
Reserved	1Bx2	8'd 214	Write once after system reset	Read out after system reset, bus readable when rom_rd_en valid		Customer trim
		8'd 215				
User data	32B	8'd 216	Write once after system reset	Read out after system reset, bus readable	User defined data For more details, refer to User data register (EFUSE_USER_DATA)	User
Reserved	8B	8'd 248	can not modify			

Note: System parameters with 'custom trim' attribute, refer to [RF data register](#)
[\(EFUSE_RF_DATA\)](#) for details.

3.4.3. Read operation

The value of the EFUSE can only be accessed through the corresponding register. After system reset, the EFUSE value take effect and reloaded to corresponding register. When read the ROTPK or its hash/Debug Password/IAK/GSSA/User data or RF data related bytes in the EFUSE, you need to follow the following steps:

The following steps show the register access sequence of the EFUSE reading operation.

1. Clear the RDIF bit if it is SET, and make sure there are no overstep boundary errors.
2. Reset the EFRW bit in EFUSE_CS.
3. Write the desired efuse address and size to EFUSE_ADDR register.
4. Set the EFSTR bit EFUSE_CS register.
5. Wait until the reading operation has been finished by checking the RDIF bit in EFUSE_CS register.
6. Read the register value corresponding to the EFUSE.

When the read operation is executed successfully, the RDIF in EFUSE_CS register is set, and an interrupt will be triggered by EFUSE if the RDIE bit in the EFUSE_CS register is set.

Note: When reading EFUSE block, the whole block should be read back first, and then the corresponding register content could be read.

3.4.4. Program operation

The value of the EFUSE can only be modified through the corresponding register.

The following steps show the register access sequence of the EFUSE writing operation.

1. Clear the PGIF bit if it is SET, and make sure there are no overstep boundary errors.
2. SET the EFRW bit in EFUSE_CS.
3. Write the desired efuse address and size to EFUSE_ADDR register.
4. Write the data to the corresponding register.
5. Set the EFSTR bit EFUSE_CS register.
6. Wait until the writing operation has been finished by checking the PGIF bit in EFUSE_CS register.

When the write operation is executed successfully, the PGIF in EFUSE_CS register is set, and an interrupt will be triggered by EFUSE if the PGIE bit in the EFUSE_CS register is set. It should be noted that the address and size of the written data must match the corresponding fuse register. If not match, OBERIF bit in the EFUSE_CS register will be set, and an interrupt will be triggered by EFUSE if the OBERIE bit in the EFUSE_CS register is set.

Note: RF data can only be written in byte, the others can be written in word.

3.5. Register definition

EFUSE secure access base address: 0x5002 2800

EFUSE non-Secure access base address: 0x4002 2800

3.5.1. Control and status register (EFUSE_CS)

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
					OVBERIC	RDIC	PGIC	Reserved	OVBERIE	RDIE	PGIE	Reserved	OVBERIF	RDIF	PGIF
					rc_w1	rc_w1	rc_w1		rw	rw	rw		r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAKSEL								Reserved					EFRW	EFSTR	
													rw	rw	

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	OVBERIC	Clear bit for overstep boundary error interrupt flag 0: No effect 1: Clear error flag
25	RDIC	Clear bit for read operation completed interrupt flag 0: No effect 1: Clear read operation completed interrupt flag
24	PGIC	Clear bit for program operation completed interrupt flag 0: No effect 1: Clear program operation completed interrupt flag
23	Reserved	Must be kept at reset value.
22	OVBERIE	Enable bit for overstep boundary error interrupt 0: Disable the overstep boundary error interrupt 1: Enable the overstep boundary error interrupt
21	RDIE	Enable bit for read operation completed interrupt 0: Disable the read operation completed interrupt 1: Enable the read operation completed interrupt
20	PGIE	Enable bit for program operation completed interrupt 0: Disable the program operation completed interrupt

		1: Enable the program operation completed interrupt
19	Reserved	Must be kept at reset value.
18	OVBERIF	Overstep boundary error flag 0: No overstep boundary error occurred 1: Overstep boundary error has occurred
17	RDIF	Read operation complete flag 0: Read EFUSE operation not completed 1: Read EFUSE operation completed
16	PGIF	Program operation completed flag 0: Program EFUSE operation not completed 1: Program EFUSE operation completed
15	IAKSEL	EFUSE_IAK_GSSA register contribute configurer 0: EFUSE_IAK_GSSA register for IAK 1: EFUSE_IAK_GSSA register for GSSA
14:2	Reserved	Must be kept at reset value.
1	EFRW	The selection of efuse operation 0: Read EFUSE 1: Write EFUSE This bit cannot be modified when the EFSTR bit is 1
0	EFSTR	Start EFUSE operation This bit is set by software and cleared by hardware 0: No effect 1: Start read or write EFUSE operation

3.5.2. Address register (EFUSE_ADDR)

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

Bits	Fields	Descriptions
------	--------	--------------

31:15	Reserved	Must be kept at reset value.
14:8	EFSIZE[6:0]	Read or write EFUSE data size
7:0	EFADDR[7:0]	Read or write EFUSE data start address

Note: This register cannot be modified when the EFSTR bit is 1.

3.5.3. Control register (EFUSE_CTL)

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
Reserved								SWBOOT	EFBOOT	SWBOOT	EFBOOT	EFBOOT	EFBOOT	LK	EFSB

rw						
----	----	----	----	----	----	----

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	SWBOOT0	Efuse boot 0 bit enable 0: Select boot0_pad as BOOT0 output 1: Select efuse_boot0 as BOOT0 output
4	EFBOOT0	EFUSE boot0 0: Efuse_boot0 = 0 1: Efuse_boot0 = 1
3	SWBOOT1	EFUSE boot 1 bit enable 0: Select boot1_pad as BOOT1 output 1: Select efuse_boot1 as BOOT1 output
2	EFBOOT1	EFUSE boot1 0: Efuse_boot1 = 0 1: Efuse_boot1 = 1
1	EFBOOTLK	EFUSE_CTL register bit[5:2] lock bit 0: Unlock EFUSE_CTL register bit[5:2], these bits can be modify 1: Lock EFUSE_CTL register bit[5:2], these bits can not be modify
0	EFSB	Boot from Secure boot

This bit needs to work with the bits[5:1] and GSSACDM in SYSCFG_GSSA CMDR register to decide how to startup

0: Startup from Flash

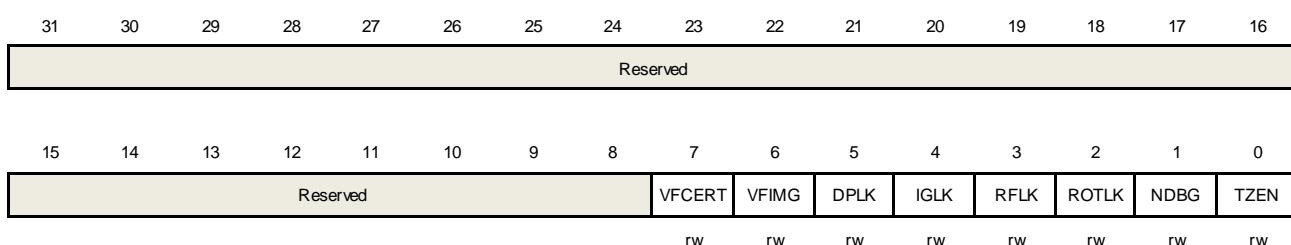
1: Startup from Secure boot

3.5.4. Trustzone control register(EFUSE_TZCTL)

Address offset: 0x0C

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	VFCERT	Verify firmware certificate 0: Disable firmware certificate verification 1: Enable firmware certificate verification
6	VFIMG	Verify firmware image 0: Disable firmware image verification 1: Enable firmware image verification
5	DPLK	EFUSE_DP register lock bit. 0: Unlock EFUSE_DP register 1: Lock EFUSE_DP register
4	IGLK	EFUSE_IAK_GSSA register lock bit 0: Unlock EFUSE_IAK_GSSA register 1: Lock EFUSE_IAK_GSSA register
3	RFLK	EFUSE_RF_DATA register lock bit 0: Unlock EFUSE_RF_DATA register, and can be modified 1: Lock EFUSE_RF_DATA register, and can not be modified
2	ROTLK	EFUSE_ROTKEY register lock bit 0: Unlock EFUSE_ROTKEY register 1: Lock EFUSE_ROTKEY register
1	NDBG	Debugging permission setting 0: Unlimited debugging

1: Can not debug

0	TZEN	Trust zone enable bit 0: Disable trust zone function 1: Enable trust zone function
---	------	--

3.5.5. Flash protection control register (EFUSE_FP_CTL)

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FP[7:0]	Efuse flash protection value Bit[7:3]: Reserved Bit2: 0~32K write protection Bit1: Read protection level 0.5 Bit0: Read protection level 1

3.5.6. User control register (EFUSE_USER_CTL)

Address offset: 0x14

Reset value: 0x0000 0006

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								UDLK AESEN MCUIINITL EF_OPLK NRSTDPS NRSTSTD HWDG							
rw								rw rw rw rw rw rw rw rw							

Bits	Fields	Descriptions
------	--------	--------------

31:6	Reserved	Must be kept at reset value.
6	UDLK	EFUSE_USER_DATA register lock bit 0: Unlock EFUSE_USER_DATA register 1: Lock EFUSE_USER_DATA register
5	AESEN	Lock EFUSE_AES_KEY register and enable AES decrypt function 0: Disable AES decrypt and AES key can be written 1: Enable AES decrypt and AES key can't be written
4	MCUINITLK	EFUSE MCU_INIT_DATA register lock bit 0: MCU initialization parameters can be written when parameters have never been written 1: MCU initialization parameters can not be written
3	EFOPLK	EFUSE_FP_CTL and EFUSE_USER_CTL register lock bit 0: Unlock EFUSE_FP_CTL and EFUSE_USER_CTL register, these bytes can be modified when parameters have never been written 1: Lock EFUSE_FP_CTL and EFUSE_USER_CTL register, these bytes can not be modified
2	NRSTDPSLP	Reset option of entry deep sleep mode 0: Generate a reset instead of entering Deep-sleep mode 1: No reset when entering Deep-sleep mode
1	NRSTSTDBY	Reset option of entry standby mode 0: Generate a reset instead of entering standby mode 1: No reset when entering standby mode
0	HWDG	Free watchdog timer selection 0: Hardware free watchdog timer 1: Software free watchdog timer

3.5.7. MCU initialization data register (EFUSE_MCU_INIT_DATA)

Address offset: 0x18+X*4(X=0,1,2)

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
INITDATA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITDATA [15:0]															
rw															

Bits	Fields	Descriptions
31:0	INITDA TA[31:0]	Efuse mcu_init value.

3.5.8. Firmware AES key register (EFUSE_AES_KEY)

Address offset: 0x24+X*4(X=0,1,2,3)

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
AESKEY[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AESKEY[15:0]															
rw															

Bits	Fields	Descriptions
31:0	AESKEY[31:0]	EFUSE AES key value.

3.5.9. RoTPK key register (EFUSE_ROT PK_KEY)

Address offset: 0x34+X*4(X=0,1,2,3,...,7)

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
RKEY[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RKEY[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RKEY[31:0]	EFUSE RoTPK or its hash value.

3.5.10. Debug password register (EFUSE_DP)

Address offset: 0x54+X*4(X=0,1)

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

DP[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DP[31:0]	EFUSE Debug password value.

3.5.11. IAK key or GSSA register (EFUSE_IAK_GSSA)

Address offset: 0x5C+X*4(X=0,1,2,3,...,15)

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
IAKGSSA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAKGSSA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IAKGSSA[31:0]	EFUSE IAK/GSSA value. The properties of this register depend on the IAKSEL bit in the EFUSE_CS register

3.5.12. Product UID register (EFUSE_PUID)

Address offset: 0x9C+X*4(X=0,1,2,3)

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
UID[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[15:0]															
r															

Bits	Fields	Descriptions
31:0	UID[31:0]	EFUSE MCU UID value.

3.5.13. HUK key register (EFUSE_HUK_KEY)

Address offset: 0xAC+X*4(X=0,1,2,3)

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
HKEY[31:16]															
									r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HKEY[15:0]															
									r						

Bits	Fields	Descriptions
31:0	HKEY[31:0]	EFUSE HUK value.

3.5.14. RF data register (EFUSE_RF_DATA)

Address offset: 0xBC+X*4(X=0,1,2,3,...,11)

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
RFDATA[31:16]															
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFDATA[15:0]															
									rw						

Bits	Fields	Descriptions
31:0	RFDATA[31:0]	EFUSE RF data value.

3.5.15. User data register (EFUSE_USER_DATA)

Address offset: 0xEC+X*4(X=0,1,2,3,...,7)

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
USERDATA[31:16]															
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERDATA[15:0]															

USERDATA[15:0]

rw

Bits	Fields	Descriptions
31:0	USERDATAA[31:0]	EFUSE USER_DATA value.

3.5.16. EFUSE Pre-TrustZone enable register (EFUSE_PRE_TZEN)

Address offset: 0x118

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

rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	STZEN	Enable Trustzone function by software. This bit will not be written to the efuse, thus it can rollback to 0. 0: Disable trust zone function 1: Enable trust zone function

3.5.17. TrustZone boot address register (EFUSE_TZ_BOOT_ADDR)

Address offset: 0x120

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

r

TZBOOTADDR[15:0]

r

Bits	Fields	Descriptions
31:0	TZBOOTADDR[31:0]	Boot from the address when TrustZone is enabled.

3.5.18. No-TrustZone boot address register (EFUSE_NTZ_BOOT_ADDR)

Address offset: 0x124

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:0	NTZBOOTADDR[31:0]	Boot from the address when TrustZone is disabled.

4. Instruction cache (ICACHE)

4.1. Introduction

The instruction cache (ICACHE) is based on C-AHB code bus of Cortex-M33 processor. It is necessary to improve performance in fetching instruction and data from both internal and external memories.

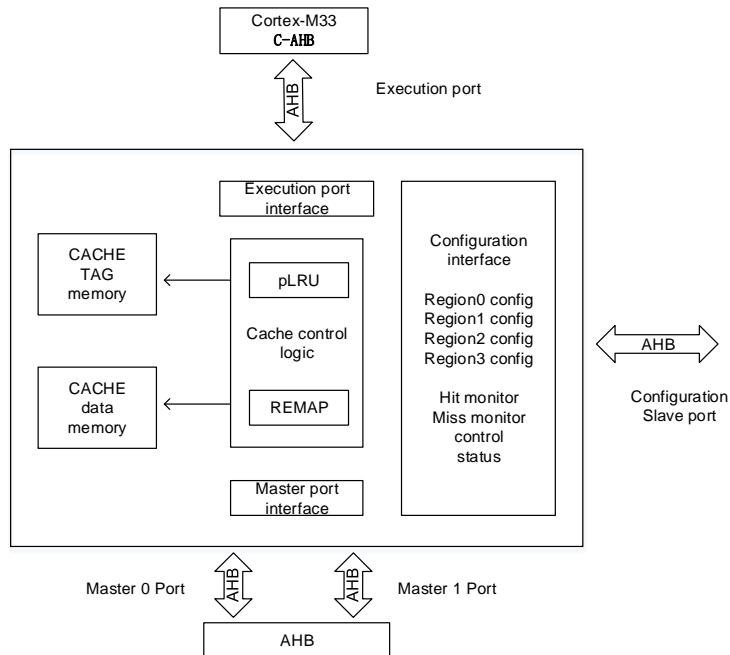
4.2. Characteristics

- Support 32KB cache with 2 ways, 1024 cache lines per way and 16B per cache line
- Support fetch address without any wait state if cache hit
- ICACHE interface support two 32-bit slave ports and two 32-bit master ports. One slave port is execution port, the other one is for registers access. Master 0 port outputs to fast bus, while master 1 port outputs to slow bus.
- Cache access support hit-under-miss and paired master access feature
- Support memory address remap
- Support pLRU replacement policy and critical-word-first refill policy.
- Support remap region AHB transaction burst type configuration.
- Support two performance counters: 32-bit hit monitor counter and 16-bit miss monitor counter.
- Operation management: Cache invalidate and optional interrupt handle.
- Support TrustZone security and configure registers to be protected at system level.

4.3. Function overview

The purpose of ICACHE is to cache fetched instruction and data, only for read transaction, not for write transaction. Considering error management, in case of detecting an unexpected cacheable write access, set an error flag and trigger optionally an interrupt.

Figure 4-1. ICACHE block diagram



4.3.1. ICACHE initialization

Set the EN bit of ICACHE_CTL register, then ICACHE function is enabled, in the same clock period. While, if ICACHE is disabled, means that ICACHE is ignored, ICACHE default state is disabled at boot.

Once released reset signal, cache invalidate operation is automatically started, each TAG valid bit to 0, INVAL bit is automatically cleared, and ICACHE_STAT BUSY flag will be set. When cache invalidate operation is ended, all cache line valid bit will be cleared, and BUSY flag is reset, while END flag is set.

To ensure performance, it is necessary to check if cache invalidate operation completed before enabling the ICACHE. Application necessarily check BUSY and END flag in ICACHE_STAT before enabling the ICACHE. In case that ICACHE is enabled before invalidate operation ended, during this period, on the condition that BUSY flag is set, any cache access is uncacheable actually and its accessing performance depends on the main memory.

4.3.2. Paired master cache

There is a paired AHB master ports: master0 and master1 port, which support ICACHE to classify the address access to different destination memories. Master0 port (fast bus) access internal flash and internal SRAM, and master1 port (slow bus) access external flash through QSPI interface. By programming the MSEL bit of ICACHE_CFGx, master0 port is selected for mapped access to internal memories, master 1 is selected for remapped traffic to external memories.

Paired master feature insure that the processor have a backup method to fetch from different memory. It is feasible to separate the traffic to internal flash and the traffic to internal SRAM (if it is remapped), so as to decrease the processor stalls when ICACHE is not on hit.

4.3.3. ICACHE TAG memory

The ICACHE TAG memory includes valid bits and address tags that indicate which data are contained in the cache data memory

For each cache line, there is one corresponding valid bit, which is set while the cache line is refilled. Valid bit is reset in several cases, for example, reset ICACHE, disable cache and execute ICACHE invalidate operation.

When input execution port receive cacheable transaction, its AHB address (AHB_ADDR_in, 32bit) is composed of Bits [3:0], Bits [13:4] and Bits [31:14]. Bits [3:0] is address byte offset, Bits [13:4] is address cache line index and Bits [31:14] is tag address. AHB address indicate the byte offset inside a cache line and cache line index inside a way, furthermore, check if the requested data is valid in cache.

ICACHE main parameters of TAG memory for default 2-way set associative is shown in [Table 4-1. TAG memory parameters.](#)

Table 4-1. TAG memory parameters

Parameter	Value
Cache size	32KB
Cache ways number	2
Cache line size	128-bit
Cache lines number	1024 per way
Address byte offset size	4-bit
Address way index size	10-bit
TAG address size	18-bit

4.3.4. Address remapping

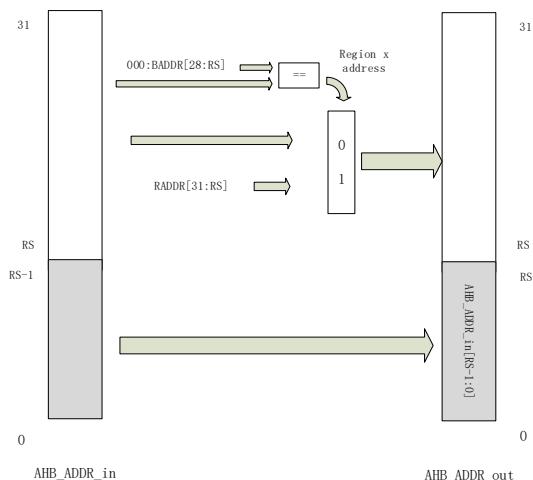
For external memory regions, it is meaningful to define an aliased address on which the address remapping apply, and transformed to the destination external physical address. The essence of remapping mechanism is mapping between input AHB address and specified code region base address.

If AHB_ADDR_in[31:RS] equals to 000:BADDR[28:RS], AHB_ADDR_in belongs to region x. BADDR is the code sub-region base address, which is defined in the BADDR field of ICACHE_CFGx. RS defines the number of available bits, it equals to log2(region size), the minimum is 21, corresponding region size is 2MB, the maximum is 27, corresponding region size is 128MB.

In case that region x is enabled, the master port output AHB address (AHB_ADDR_out)

which consist of MSBs (RADDR[31:RS] field of ICACHE_CFGx) and LSBs (AHB_ADDR_in[RS-1:0]).

Figure 4-2. ICACHE remapping address



When programming BADDR and RADDR field in ICACHE_CFGx, it is notable that if programmed value is larger than expected (MSBs), the unnecessary extra LSBs are bypassed.

Table 4-2. ICACHE remap region size

Region size (MB)	Base address (MSBs)	Remap address (MSBs)
2	8	11
4	7	10
8	6	9
16	5	8
32	4	7
64	3	6
128	2	5

The remapping operation is configured by programming ICACHE_CFGx register, the programming operation is merely done in case that ICACHE is disabled. While, once the EN bit in ICACHE_CFGx register is set, even though ICACHE is disabled, or the transaction is uncacheable, the corresponding region x is enabled and then remap operation is generated. The SIZE bit in ICACHE_CFGx register is used to configure remap region size. The size of region is a power of two multiple in 2 Mbytes, the minimum region size is 2 Mbytes and the maximum region size is 128 Mbytes.

Through configuring the OBT bit in ICACHE_CFGx, the type of AHB burst is configured as INCR4.

INCR4 is used for external memories accessed through QSPI interface. INCR4 burst size equals to cache line size. INCR4 burst start address equals to cache line boundary aligned address.

4.3.5. Cacheable and uncacheable access

ICACHE support memory region remapping feature, so as to make external memory regions cacheable. ICACHE is able to access up to four external memory regions in cacheable way. There is an alias of external memory region address in the code region (address range [0x0000 0000:0x07FF FFFF] or [0x1000 0000:0x1FFF FFFF]), so ICACHE can manage and C-AHB bus is able to route external memory region (physical address range [0x9000 0000:0x97FF FFFF]) through their code alias address.

Depending on memory request AHB transaction memory lookup attribute, which is shown in [Table 4-3. ICACHE cache ability for transaction](#), the request to ICACHE is defined as cacheable or uncacheable.

Table 4-3. ICACHE cache ability for transaction

Lookup attribute	Cache ability
1	Cacheable
0	Uncacheable

If there is a memory request for external memory in the code region currently, ICACHE operate the address remapping firstly. If the address is necessary to be aliased, remap address firstly, and then cached. The target physical address does not need further operation. The remapping functionality is remain available even if cache is disabled and traffic is uncacheable.

If there is a memory uncacheable access, ICACHE is ignored, so that the AHB transaction is transmitted directly unchanged to the master output port, while, depending on remapping feature, only transaction address may be changed. The ignored operation has no effect on accessing the target memory.

User is able to configure Cacheable memory regions in the memory protection unit (MPU), which is responsible for the AHB attribute signal generating, the attribute is serve for any transaction addressing a given region.

Table 4-4. memory configuration

memory	Cacheable (MPU Programming)	Remap (ICACHE_CFGx Programming)
Flash	Yes or No	Not required
SRAM	Not recommended	Not required
External memory (QSPI interface)	Yes or No	Required

If ICACHE receive a cacheable transaction request, there are two scenes come out: cache hit and cache miss.

Cache hit: If address is present in its TAG memory and the corresponding cache line is valid, means cache hit, reading from cache and providing to core are in the same period.

However, according to hit-under-miss feature, if address is not present in TAG memory which belongs to the current refill burst, also means cache hit. Even if cache line is refilling, only if the data is available at its master interface, ICACHE is able to fetch the requested data quickly, so as to avoid a miss.

Cache miss: If address is not present in its TAG memory, means cache miss, the data is read from main memory, and refill the cache line. The critical-word-first policy optimize wait periods for the processor, the policy allows processor to read the requested data firstly, and then, cache still performs cache line refilling. If an address remap happens, the kind of burst, which is generated on master port, is configured by the OBT bit in its ICACHE_CFGx register.

Considering cache refilling, ICACHE determines cache line index. In 2-way set associative mode, the line is pointed by the address index in each way, according to pLRU replacement algorithm, the way is selected for use, and the other one is for the next refill. In the 2-way set associative cache mode, if the cache line is selected to be written by the refill data, and the line is already valid, the targeted cache line must be invalidated first.

Comparing to fetching instructions from main memory, processor gets lower power consumption in fetching instructions from internal ICACHE. If the cached memory is external, power saving is more obvious.

4.3.6. ICACHE performance monitoring

For analyzing cache performance, ICACHE supports two monitors: a 32-bit hit monitor and a 16-bit miss monitor, they are disabled by default.

Hit monitor counts the AHB-transactions on ICACHE input port, which do not generate a transaction on output master0 or master1 port. It is necessary to consider the access whose address is present in the TAG memory or in the refill buffer.

Miss monitor counts the AHB-transactions on ICACHE input port, which generate a transaction on output master0 or master1 port. It is necessary to consider the access whose address is not present in the TAG memory and the refill buffer.

Hit and miss monitors can be enabled and reset by software as follows:

- Enable / switch off the hit monitor through the HMEN bit in ICACHE_CTL.
- Reset the hit monitor by setting the HMRST bit in ICACHE_CTL.
- Enable / switch off the miss monitor through the MMEN bit in ICACHE_CTL.
- Reset the miss monitor by setting the MMRST bit in ICACHE_CTL.

4.3.7. ICACHE interrupts

If there is an unsupported cacheable write request detected, an error is generated by setting ERR bit in ICACHE_STAT. Meanwhile, if the corresponding interrupt enable bit is set, error interrupt is triggered.

If a cache invalidation operation is finished, an end flag is generated by setting END bit

in ICACHE_STAT. meanwhile, if the corresponding interrupt enable bit is set, end interrupt is triggered, and then cache is available again.

Table 4-5. ICACHE interrupt

ICACHE	error	end
Interrupt event	Functional error	Operation end
Event flag	ERR	END
Interrupt enable bit	ERRIE	ENDIE
Interrupt clear bit	ERRC	ENDC

4.4. Register definition

ICACHE secure access base address: 0x5008 0000

ICACHE non-secure access base address: 0x4008 0000

4.4.1. Control register (ICACHE_CTL)

Address offset: 0x00

Reset value: 0x0000 0004

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										MMRST	HMRST	MMEN	HMEN		
										rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										BSTT	AMSEL	INVAL	EN		
										rw	rw	w	rw		

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	MMRST	Miss monitor reset 0: no effect 1: reset cache miss monitor
18	HMRST	Hit monitor reset 0: no effect 1: reset cache hit monitor
17	MMEN	Miss monitor enable 0: switch off cache miss monitor, stopping the monitor and does not reset. 1: cache miss monitor enabled
16	HMEN	Hit monitor enable 0: switch off cache hit monitor, stopping the monitor and does not reset. 1: cache hit monitor enabled
15:4	Reserved	Must be kept at reset value
3	BSTT	Burst type for fast bus 0: WRAP4 1: INCR4
2	AMSEL	Cache set-associativity mode selection Configure set-associativity mode when cache is disabled, software write. 0: no effect 1: 2-way set associative cache (reset value)
1	INVAL	Cache invalidation, Set by software and cleared by hardware (BUSYF flag is set) 0: no effect

		1: invalidate entire cache
0	EN	Enable
		0: cache disabled
		1: cache enabled

4.4.2. Status register (ICACHE_STAT)

Address offset: 0x04

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERR	Cache error flag 0: no error 1: error occurred during the operation
1	END	operation end flag 0: cache busy 1: invalidate INVAL operation ended
0	BUSY	Busy flag 0: cache is not executing a invalidate operation 1: cache is executing a invalidate operation

4.4.3. Interrupt enable register (ICACHE_INTEN)

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
Reserved												ERRIE	ENDIE	Reserved	
												rw	rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERRIE	cache error interrupt enable 0: disable error interrupt 1: enable error interrupt
1	ENDIE	cache operation end interrupt enable 0: disable operation end interrupt 1: enable operation end interrupt
0	Reserved	Must be kept at reset value

4.4.4. Flag clear register (ICACHE_FC)

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
Reserved												ERRC	ENDC	Reserved	
												w	w		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERRC	Softw are clear cache error flag 0: no effect 1: clears ERR flag in ICACHE_STAT
1	ENDC	Softw are clear operation end flag 0: no effect 1: clears END flag in ICACHE_STAT.
0	Reserved	Must be kept at reset value

4.4.5. Hit monitor counter register (ICACHE_HMC)

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
HMC[31:16]															

r

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

r

Bits	Fields	Descriptions
31:0	HMC [31:0]	Cache hit monitor counter.

4.4.6. Miss monitor counter register (ICACHE_MMCR)

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	MMC[15:0]	Cache miss monitor counter

4.4.7. Configuration register (ICACHE_CFGx)

Address offset: 0x20 + 4 * x, (x = 0..3)

Reset value: 0x0000 0200

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
OBT	Reserved	MSEL	Reserved					RADDR							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	Reserved			SIZE	Reserved					BADDR					
rw															

Bits	Fields	Descriptions
31	OBT	Output burst type for region x 0: Reserved 1: INCR4
30:29	Reserved	Must be kept at reset value

28	MSEL	Region x AHB cache master selection 0: select master0 by default 1: select master1
27	Reserved	Must be kept at reset value
26:16	RADDR	Region x remapped address, this field replaces the alias address defined by BADDR field.
15	EN	Region x enable 0: disabled 1: enabled
14:12	Reserved	Must be kept at reset value
11:9	SIZE	Region x size 000: reserved 001: 2 Mbytes 010: 4 Mbytes 011: 8 Mbytes 100: 16 Mbytes 101: 32 Mbytes 110: 64 Mbytes 111: 128 Mbytes
8	Reserved	Must be kept at reset value
7:0	BADDR	Region x base address

5. Power management unit (PMU)

5.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32W51x series. According to the Power management unit (PMU), provides five types of power saving modes, including Sleep, Deep-sleep, Standby, SRAM_sleep and Wi-Fi_sleep mode. These modes reduce the power consumption and allow the application to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For GD32W51x devices, there are three power domains, including V_{DD} / V_{DDA} domain, 1.2V domain, and Backup domain, as is shown in the following figure. 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 1.2V domain power. A powerswitch is implemented for the Backup domain. It can be powered from the V_{BAT} voltage when the main V_{DD} supply is shut down.

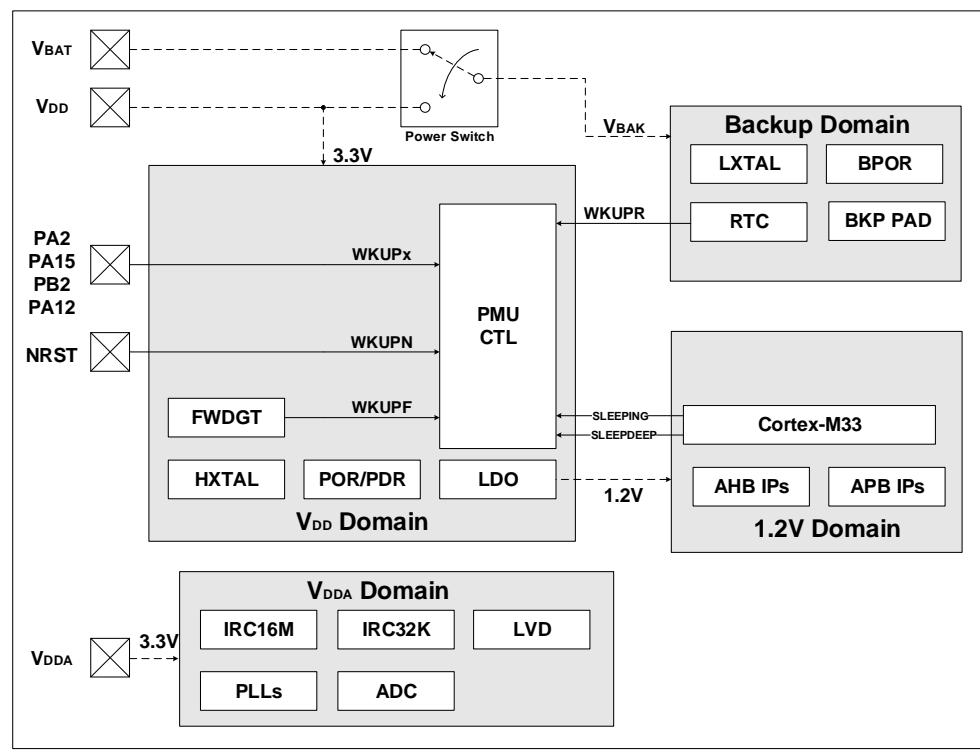
5.2. Characteristics

- Three power domains: V_{BAK} , V_{DD} / V_{DDA} and 1.2V power domains.
- Five power saving modes: Sleep, Deep-sleep, Standby, SRAM_sleep and Wi-Fi_sleep modes.
- Internal Voltage regulator (LDO) supplies around 1.1V / 1.2V voltage source for 1.2V domain.
- V_{DD} Low Voltage Detector can issue an interrupt or event when the power is lower than a programmed threshold.
- V_{DDA} Low Voltage Detector can issue an interrupt or event when the V_{DDA} is equal to or lower than the specified threshold (2.4V).
- Battery power (V_{BAT}) for Backup domain when V_{DD} is shut down.
- LDO output voltage select for power saving.
- Ultra power saving for low-driver mode in Deep-sleep mode.
- SRAM1 / SRAM2 / SRAM3 can be power-off
- Wi-Fi_OFF domain can be power-off

5.3. Function overview

[Figure 4-3. Power supply overview](#) provides details on the internal configuration of the PMU and the relevant power domains.

Figure 4-3. Power supply overview



VRFx: RF Power

5.3.1. Battery backup domain

The Backup domain is powered by the V_{DD} or the battery power source (V_{BAT}) selected by the internal power switch, and the V_{BAK} pin which drives Backup Domain, power supply for RTC unit, LXTAL oscillator, BPOR, and two pads, including PC14 to PC15. In order to ensure the content of the Backup domain registers and the RTC supply, when V_{DD} supply is shut down, V_{BAT} pin can be connected to an optional standby voltage supplied by a battery or by another source. The power switch is controlled by the Power Down Reset circuit in the V_{DD} / V_{DDA} domain. If no external battery is used in the application, it is recommended to connect V_{BAT} pin externally to V_{DD} pin with a 100nF external ceramic decoupling capacitor.

The Backup domain reset sources includes the Backup domain power-on-reset (BPOR) and the Backup Domain software reset. The BPOR signal forces the device to stay in the reset mode until V_{BAK} is completely powered up. Also the application software can trigger the Backup domain software reset by setting the BKPRST bit in the RCU_BDCTL register to reset the Backup domain.

The clock source of the Real Time Clock (RTC) circuit can be derived from the Internal 32KHz RC oscillator (IRC32K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 2 to 31. When V_{DD} is shut down, only LXTAL is valid for RTC. Before entering the power saving mode by executing the WFI / WFE instruction, the Cortex®-M33 can setup the RTC register with an expected alarm time and enable the alarm function and according EXTI lines to achieve the RTC alarm event. After entering the power saving mode for a certain amount of time, the RTC alarm will wake up the device when the time match event occurs. The details of the RTC configuration and operation will be described in the [Real time clock \(RTC\)](#).

When the Backup domain is supplied by V_{DD} (V_{BAK} pin is connected to V_{DD}), the following function is available:

- PC14 and PC15 can be used as either GPIO or LXTAL Crystal oscillator pins.

When the Backup domain is supplied by V_{BAT} (V_{BAK} pin is connected to V_{BAT}), the following function is available:

- PC14 and PC15 can be used as LXTAL Crystal oscillator pins only.

Note: Since PC14 and PC15 are supplied through the Power Switch, which can only be obtained by a small current, the speed of GPIOs PC14 to PC15 should not exceed 2MHz when they are in output mode (maximum load: 30pF).

5.3.2. VDD / VDDA power domain

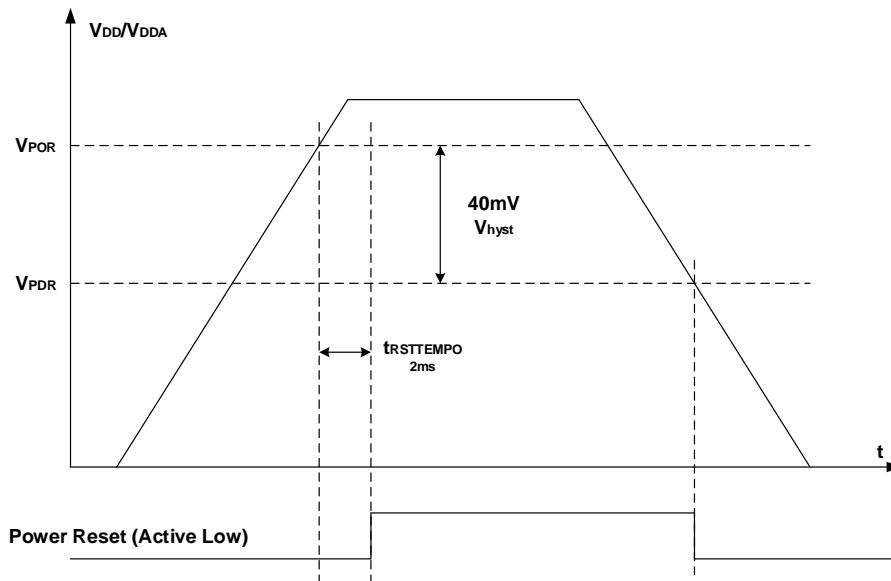
V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes HXTAL (High Speed Crystal oscillator), LDO (Voltage Regulator), POR / PDR (Power On / Down Reset), FWDGT (Free Watchdog Timer), all pads except PC14 / PC15, etc. V_{DDA} domain includes ADC (AD Converter), IRC16M (Internal 16MHz RC oscillator), IRC32K (Internal 32KHz RC oscillator), PLLs (Phase Locking Loop), LVD (Low Voltage Detector), etc.

V_{DD} domain

The LDO, which is implemented to supply power for the 1.2V domain, is always enabled after reset. It can be configured to operate in three different status, including in the Sleep mode (full power on), in the Deep-sleep mode (on or low power), and in the Standby mode (power off).

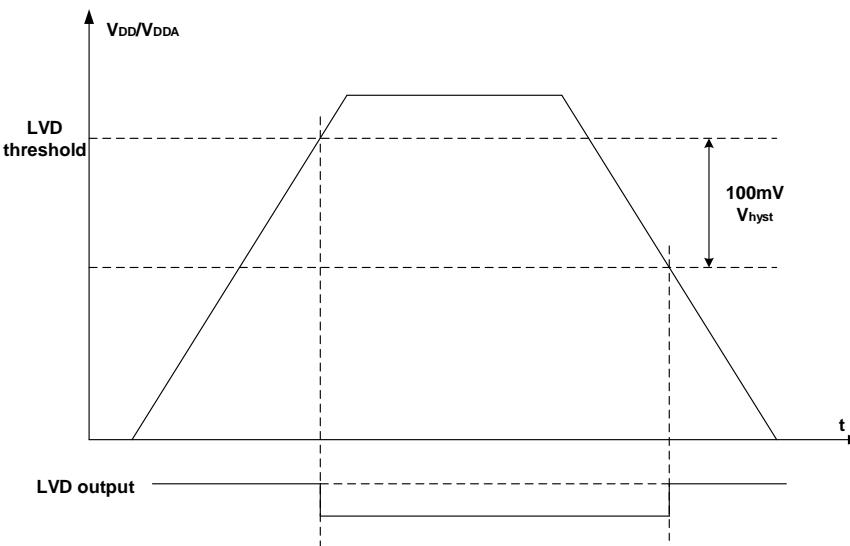
The POR / PDR circuit is implemented to detect V_{DD} / V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold. [Figure 4-4. Waveform of the POR / PDR](#) shows the relationship between the supply voltage and the power reset signal. V_{POR} , which typical value is 1.54V, indicates the threshold of power on reset, while V_{PDR} , which typical value is 1.50V, means the threshold of power down reset. The hysteresis voltage (V_{hyst}) is around 40mV.

Figure 4-4. Waveform of the POR / PDR



The LVD is used to detect whether the V_{DD} supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the Power control register (PMU_CTL0). The LVD is enabled by setting the LVDEN bit, and LVDF bit, which in the PMU_CS0 register, indicates if V_{DD} is higher or lower than the LVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. [Figure 4-5](#). [Waveform of the LVD threshold](#) shows the relationship between the LVD threshold and the LVD output (LVD interrupt signal depends on EXTI line 16 rising or falling edge configuration). The following figure shows the relationship between the supply voltage and the LVD signal. The hysteresis voltage (V_{hyst}) is 100mV.

Figure 4-5. Waveform of the LVD threshold



V_{DDA} domain

The VLVD (V_{DDA} Low Voltage Detector) is used to detect whether the V_{DDA} is equal to or lower

than the specified threshold (2.4V). The VLVD is enabled by setting the VLVDEN bit in PMU_CTL0 register, and VLVD bit, which in the PMU_CS0 register, indicates if V_{DDA} is higher or lower than the specified threshold (2.4V). This event is internally connected to the EXTI line 18 and can generate an interrupt if it is enabled through the EXTI registers. VLVD bit interrupt signal depends on EXTI line 18 rising or falling edge configuration.

Generally, digital circuits are powered by V_{DD} , while most of analog circuits are powered by V_{DDA} . To improve the ADC 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.

5.3.3. 1.2V power domain

The main functions that include Cortex®-M33 logic, AHB / APB peripherals, the APB interfaces for the Backup domain and the V_{DD} / V_{DDA} domain, etc, are located in this power domain. Once the 1.2V is powered up, the POR will generate a reset sequence on the 1.2V power domain. If need to enter the expected 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 an expected power saving mode which will be discussed in the following section.

CORE_MEM1/2/3 domain

In GD32W51x, the CORE_MEM1/2/3 power domain is defined for 64KB SRAM1 / 128KB SRAM2 / 192KB SRAM3 respectively. When run / sleep / deep-sleep mode, the SRAMs can power-off. The typical work state is as follows:

1. Power off when standby mode / BKP_ONLY mode.
2. Power on (default) when controlled by register in run / sleep / deep_sleep mode.
3. Power off when controlled by register in run / sleep / deep_sleep mode.

Wi-Fi_OFF domain

Refer to Wi-Fi spec. The typical work state is as follows:

1. Power off when standby mode / BKP_ONLY mode.
2. Power on when controlled by register in run / sleep / deep_sleep mode.
3. Power off (default) when controlled by register in run / sleep / deep_sleep mode.

Note:

BKP_ONLY mode: Enter when V_{DD} pin cut off by external Power switch, while the V_{BAT} pin supply.

5.3.4. Power saving modes

After a system reset or a power reset, the GD32W51x MCU operates at full function and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1, and PCLK2) or gating the clocks of the unused peripherals or configuring the LDO output voltage by LDOVS bits in PMU_CTL0 register. The LDOVS bits should be configured only when the PLL is off. Besides, five power saving modes are provided to achieve even lower power consumption, they are Sleep mode, Deep-sleep mode, Standby, SRAM_sleep and Wi-Fi_sleep 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 offers 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 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 lowest priority ISR.

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 1.2V domain are off, and all of IRC16M, HXTAL and PLLs are disabled. The contents of SRAM0 and registers are preserved. The LDO can operate normally or in low power mode depending on the LDOLP bit in the PMU_CTL0 register. 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_CTL0 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 IRC16M is selected as the system clock. Notice that an additional wakeup delay will be incurred if the LDO operates in low power mode.

The low-driver mode in Deep-sleep mode can be entered by configuring the LDEN, LDNP,

LDLP, LDOLP bits in the PMU_CTL0 register. The low-driver mode provides lower drive capability, and the low-power mode take lower power.

Normal-driver & Normal-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL0 register, and not in low-power mode depending on the LDOLP bit reset in the PMU_CTL0 register.

Normal-driver & Low-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL0 register. The low-power mode enters depending on the LDOLP bit set in the PMU_CTL0 register.

Low-driver & Normal-power: The low-driver mode in Deep-sleep mode when the LDO in normal-power mode depending on the LDOLP bit reset in the PMU_CTL0 register enters by configure LDEN to 0b11 and LDNP to 1 in the PMU_CTL0 register.

Low-driver & Low-power: The low-driver mode in Deep-sleep mode when the LDO in low-power mode depending on the LDOLP bit set in the PMU_CTL0 register enters by configure LDEN to 0b11 and LDLP to 1 in the PMU_CTL0 register.

No Low-driver: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL0 register.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and RTC Alarm / timestamp / tamper / auto wakeup flag must be reset. 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 based on the SLEEPDEEP mode of the Cortex®-M33, too. In Standby mode, the whole 1.2V domain is power off, the LDO is shut down, and all of IRC16M, HXTAL and PLLs are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and set the STBMOD bit in the PMU_CTL0 register, and clear WUF bit in the PMU_CS0 register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS0 register indicates that the MCU has been in Standby mode. There are four wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm / time stamp / tamper / auto wakeup events, the FWDGT reset, and the rising edge on WKUP pins. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM0 / SRAM1 / SRAM2 / SRAM3 and registers in 1.2V power 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 0x00000000 address.

SRAM_sleep mode

If at least one of SRAM1 / SRAM2 / SRAM3 is powered off, the SRAM enters SRAM_sleep mode. When the SRAMxPSLEEP (x = 1/2/3) bit in PMU_CTL1 register is set, the SRAMx (x = 1/2/3) will be powered off, the contents of SRAMx (x = 1/2/3) will lost. When the

SRAM_xPWAKE (x = 1/2/3) bit in PMU_CTL1 register is set, the SRAM_x (x = 1/2/3) will be powered on.

SRAM1 / SRAM2 / SRAM3 can be configured power on or power off when in run / sleep / deep_sleep mode.

SRAM1 / SRAM2 / SRAM3 are power off when in standby mode / BKP_ONLY mode.

Wi-Fi_sleep mode

The Wi-Fi_sleep mode can enter by software (set WPEN bit to 1 and set WPSLEEP bit to 1), or by hardware (driven by Wi-Fi hardware signal sleep_wl when WPEN is 1). This mode can exit by clearing WPEN bit to 0, or by setting WPEN bit to 1 then setting WPSLEEP bit to 1, or by hardware (driven by Wi-Fi hardware signal wake_wl when WPEN is 1).

When Wi-Fi enter Wi-Fi_sleep mode, Wi-Fi_OFF domain power off.

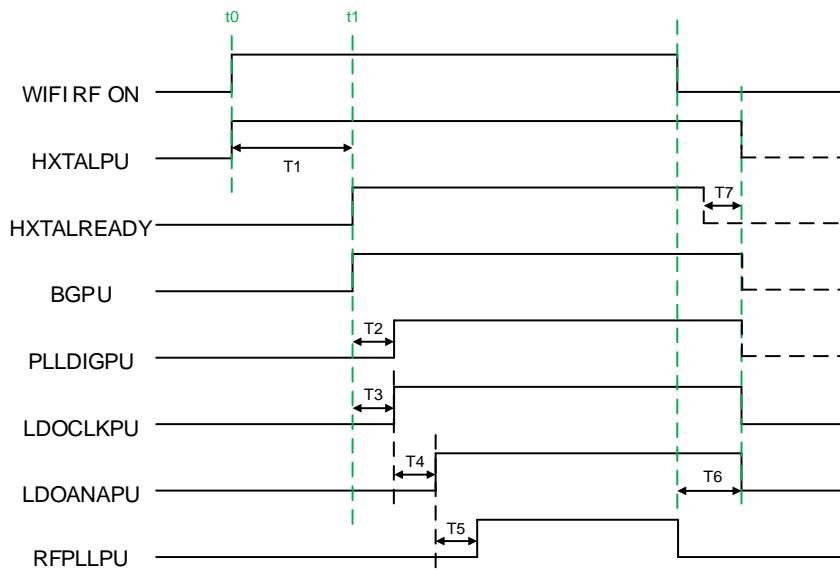
When exit from Wi-Fi_sleep mode, Wi-Fi is active mode, all Wi-Fi power domain power on.

Table 4-6 Typical work mode

Power mode	MCU	Wi-Fi	Consumption	Description
Wi-Fi tx	Run	TX	451mA	All work
Wi-Fi rx	Run	RX	127mA	All work
Modem sleep	Run	Disable	30~69mA	Wi-Fi_OFF power off, RF power off
Sensor hub	Sleep + SRAM_sleep	Disable	100~800uA	MCU sleep, CORE_MEM1/2/3 power off, Wi-Fi_OFF power off, RF power off
Light sleep	Deep-sleep + SRAM_sleep	Connective IDLE	598+50uA	MCU deep-sleep, CORE_MEM1/2/3 power off, Wi-Fi_OFF and RF off-on switch
Deeper sleep	Deep-sleep + SRAM_sleep	Disable	62+20uA	MCU deep-sleep, CORE_MEM1/2/3 power off, Wi-Fi_OFF power off, RF power off
Standby	Standby	Disable	6uA	-
Battery	BKP_ONLY	Disable	1uA	V _{BAT} only

The RF sequence is the interface of RF module, the [**Figure 4-6. RF sequence**](#) shows the RF sequence.

Figure 4-6. RF sequence



HXTALPU, HXTALREADY and PLLDIGPU are bits of RCU_CTL register; BGPU, LDOCLKPU, LDOANAPU and RFPLLPU are bits of RCU_CFG1 register.

When PLLDIGPU is 1, PLLDIGOSEL[1:0] bits in RCU_PLLCFG register should not change, and no rising edge should appear on PLLDIGEN bit of RCU_CTL register.

- If RFSWEN bit is 0, choose hardware mode to configure RF sequence:

When set WPSLEEP bit to 1, or Wi-Fi hardware signal sleep_wl is valid, RFPLL / XTAL / BG / RF ANA clocks will be automatically closed according to [Table 4-7. Time in RF sequence](#) value in order of [Figure 4-6. RF sequence](#);

When set WPWAKE bit to 1, or Wi-Fi hardware signal wake_wl is valid, RFPLL / XTAL / BG / RF ANA clocks will be automatically opened according to [Table 4-7. Time in RF sequence](#) value in order of [Figure 4-6. RF sequence](#).

- If RFSWEN bit is 1, choose software mode to configure RF sequence:

RFPLL / XTAL / BG / RFANA clocks will be opened or closed by configuring RCU related registers (recommend to configure in order of [Figure 4-6. RF sequence](#). If related registers are not configured, the clocks will remain as before).

$$t1 = t0 + T1 \quad (5-1)$$

Table 4-7. Time in RF sequence

Name	Time	Description
T1	HXTAL mode: 1ms. External supply mode (HXTAL bypass mode): 1us. (Note that the external supply clock is already ready)	HXTAL ready time
T2	1us	BandGap start time
T3	1us	-
T4	1us	Pow er up interval

Name	Time	Description
T5	1us	Power up interval
T6	1us	-
T7	1us	The reserved time of simulated closing output clock of HXTAL

If HXTAL is selected to output clock, when HXTALREADY and HXTALEN in RCU_CTL register are 1, the output clock will be stable within a HXTAL period.

Table 4-8. Power saving mode summary

Mode	Sleep	Deep-sleep	Standby	SRAM_sleep	Wi-Fi_sleep
Description	Only CPU clock is off	All clocks in the 1.2V domain are off Disable IRC16M, HXTAL and PLLs	The 1.2V domain is power off Disable IRC16M, HXTAL and PLLs	at least one of SRAM1 / SRAM2 / SRAM3 is power off	Wi-Fi_OFF is power off
LDO Status	On	On or in low power mode or low-driver mode	Off	On or in low power mode or low-driver mode	On or in low power mode or low-driver mode
Configuration	SLEEPDEE P = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST = 1	SRAMxPSLEE P = 1 (x = 1/2/3)	1. WPEN = 1, WPSLEEP = 1 2. Or hardware signal sleep_wl valid when WPEN = 1
Entry	WFI or WFE	WFI or WFE	WFI or WFE	-	-
Wakeup	Any interrupt for WFI Any event (or interrupt when SEVONPEND D is 1) for WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	NRST pin WKUP pins FWDGT reset RTC	SRAMxPWAKE = 1(x = 1/2/3)	1. WPWAKE = 1 when WPEN = 1 2. Or clear WPEN = 0 3. Or hardware signal wake_wl valid when WPEN = 1
Wakeup Latency	None	IRC16M wakeup time, LDO wakeup time added if LDO is in low power mode	Power on sequence	100ns	100ns

5.3.5. Security description

PMU secure protection

PMU register bits can be configured secured by PMU_SECCFG register against non-secure access when the TZEN bit in the EFUSE_TZCTL register is 1. LDOVS[1:0] bits can be secured when the system clock selection is secure in RCU. The features can be secured is shown as following:

- Power saving modes
- Wakeup (WKUPx) pins
- V_{DD} / V_{DDA} voltage detection
- Backup write access
- RF secure
- LDO output voltage selection

Table 4-9. PMU Security configuration summary

Security configuration bit	Secured bits	Non-secure access on secure bits
NA ⁽¹⁾	All bits in PMU_SECCFG register	Read is OK. WI and illegal access event
at least one bit in PMU_SECCFG register is set	PRIV bit in PMU_PRICFG register	Read is OK. WI and illegal access event ⁽²⁾
LPMSEC in PMU_SECCFG register is set	LDOLP, STBMOD, WURST, and STBRST bits in PMU_CTL0 register	RAZ / WI
LPMSEC or WUPxSEC in PMU_SECCFG register is set	WUPENx bits in PMU_CS0 register	RAZ / WI
VDMSEC in PMU_SECCFG register is set	LVDEN, LVDT[2:0], and VLVDEN bits in PMU_CTL0 register	RAZ / WI
DBPSEC in PMU_SECCFG register is set	BKPWEN bit in PMU_CTL0 register	RAZ / WI
LPSSEC in PMU_SECCFG register is set	All bits in PMU_CTL1 register	RAZ / WI
RFSEC in PMU_SECCFG register is set	All bits in PMU_RFCTL register	RAZ / WI
SYSCLKSEC in RCU_SECCFG register is set	LDOVS[1:0] bits in PMU_CTL0 register	RAZ / WI

Note:

1. PMU_SECCFG register is always secure.
2. Illegal access event is generated only when the PMU_PRICFG is secure.

PMU privilege protection

PRIV bit in PMU_PRICFG register determines the whether the unprivileged access to PMU

registers is supported or not. When PRIV bit is reset, both privileged and unprivileged accesses to PMU registers are supported. When PRIV bit is set, only privileged access to PMU registers is supported, unprivileged access to a privileged register is RAZ / WI.

5.4. Register definition

PMU secure access base address: 0x5000 7000

PMU non-secure access base address: 0x4000 7000

5.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x0000 C000 (reset by wakeup from Standby mode)

A non-secure read / write access on secured bits is RAZ / WI. When TZEN = 0, there is no access restriction. When PRIV in PMU_PRICFG register is 1, only privileged access is supported.

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												LDEN[1:0]	Reserved		
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDOVS[1:0]	Reserved	LDNP	LDLP	VLDEN	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP		
rs		rw	rw	rw	rw	rw			rw	rc_w1	rc_w1	rw	rw		

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:18	LDEN[1:0]	Low -driver mode enable in Deep-sleep mode 00: Low -driver mode disable in Deep-sleep mode 01: Reserved 10: Reserved 11: Low -driver mode enable in Deep-sleep mode
17:16	Reserved	Must be kept at reset value.
15:14	LDOVS[1:0]	LDO output voltage select These bits are set by software when the main PLL closed. And the LDO output voltage selected by LDOVS bits takes effect when the main PLL enabled. If the main PLL closed, the LDO output voltage low mode selected. 0x: LDO output voltage low mode (1.1V) 1x: LDO output voltage high mode (1.2V)
13:12	Reserved	Must be kept at reset value.
11	LDNP	Low -driver mode when use normal power LDO 0: normal driver when use normal power LDO 1: Low -driver mode enabled when LDEN is 11 and use normal power LDO

10	LDLP	Low -driver mode when use low power LDO. 0: normal driver when use low power LDO 1: Low -driver mode enabled when LDEN is 11 and use low power LDO
9	VLVDEN	V_{DDA} low voltage detect enable 0 : No detect 1 : Enable V_{DDA} low voltage detect
8	BKPWEN	Backup Domain Write Enable 0: Disable write access to the registers in Backup domain 1: Enable write access to the registers in Backup domain After reset, any write access to the registers in Backup domain is ignored. This bit has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low Voltage Detector Threshold 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: 3.1V
4	LVDEN	Low Voltage Detector Enable 0: Disable Low Voltage Detector 1: Enable Low Voltage Detector
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	LDOLP	LDO Low Power Mode 0: The LDO operates normally during the Deep-sleep mode 1: The LDO is in low power mode during the Deep-sleep mode

5.4.2. Control and status register 0 (PMU_CS0)

Address offset: 0x04

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

When PRIV in PMU_PRICFG register is 1, only privileged access is supported.

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										LDRF[1:0]		Reserved				
<i>rc_w1</i>																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	LDOVSR F	Reserved	WUPEN3	WUPEN2	WUPEN1	WUPEN0	Reserved	Reserved	VLVDF	LVDF	STBF	WUF	r	r	r	r
	r		rw	rw	rw	rw			r	r	r	r				

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:18	LDRF[1:0]	<p>Low -driver mode ready flag</p> <p>These bits are set by hardware when enter Deep-sleep mode and the LDO in Low - driver mode. These bits are cleared by software when write 11.</p> <p>00: normal driver in Deep-sleep mode</p> <p>01: Reserved</p> <p>10: Reserved</p> <p>11: Low -driver mode in Deep-sleep mode</p>
17:15	Reserved	Must be kept at reset value.
14	LDOVSRF	<p>LDO voltage select ready flag</p> <p>0: LDO voltage select not ready</p> <p>1: LDO voltage select ready</p>
13:12	Reserved	Must be kept at reset value.
11	WUPEN3	<p>WKUP Pin3 (PA12) Enable</p> <p>0: Disable WKUP pin3 function</p> <p>1: Enable WKUP pin3 function</p> <p>If WUPEN3 is set before entering the power saving mode, a rising edge on the WKUP pin3 wakes up the system from the power saving mode. As the WKUP pin3 is active high, the WKUP pin3 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.</p>
10	WUPEN2	<p>WKUP Pin2 (PB2) Enable</p> <p>0: Disable WKUP pin2 function</p> <p>1: Enable WKUP pin2 function</p>

If WUPEN2 is set before entering the power saving mode, a rising edge on the WKUP pin2 wakes up the system from the power saving mode. As the WKUP pin2 is active high, the WKUP pin2 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.

9	WUPEN1	WKUP Pin1 (PA15) Enable 0: Disable WKUP pin1 function 1: Enable WKUP pin1 function If WUPEN1 is set before entering the power saving mode, a rising edge on the WKUP pin1 wakes up the system from the power saving mode. As the WKUP pin1 is active high, the WKUP pin1 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
8	WUPEN0	WKUP Pin0 (PA2) Enable 0: Disable WKUP pin0 function 1: Enable WKUP pin0 function If WUPEN0 is set before entering the power saving mode, a rising edge on the WKUP pin0 wakes up the system from the power saving mode. As the WKUP pin0 is active high, the WKUP pin0 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
7:4	Reserved	Must be kept at reset value.
3	VLVDF	V _{DDA} Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V _{DDA} is higher than the specified threshold (2.4V)) 1: Low Voltage event occurred (V _{DDA} is equal to or lower than the specified threshold (2.4V)) Note: The VLVD function is stopped in Standby mode.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V _{DD} is higher than the specified LVD threshold) 1: Low Voltage event occurred (V _{DD} is equal to or lower than the specified LVD threshold) Note: The LVD function is stopped in Standby mode.
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_CTL0 register.
0	WUF	Wakeup Flag 0: No wakeup event has been received 1: Wakeup event occurred from the WKUP pins or the RTC wakeup event including RTC Tamper event, RTC alarm event, RTC Time Stamp event or RTC Wakeup

This bit is cleared only by a POR / PDR or by setting the WURST bit in the PMU CTL0 register.

5.4.3. Control register 1 (PMU_CTL1)

Address offset: 0x08

Reset value: 0x0000 0002 (reset by wakeup from Standby mode)

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	SRAM3P WAKE	SRAM3PS LEEP	Reserved	SRAM2P WAKE	SRAM2PS LEEP	Reserved	SRAM1P WAKE	SRAM1PS LEEP	Reserved	WPWAKE	WPSLEEP	WPEN	Reserved	FW	FW

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	SRAM3P_WAKE	SRAM3 wakeup by software. Clear by hardware
13	SRAM3PSLEEP	SRAM3 go to sleep by software. Clear by hardware
12:11	Reserved	Must be kept at reset value.
10	SRAM2P_WAKE	SRAM2 wakeup by software. Clear by hardware
9	SRAM2PSLEEP	SRAM2 go to sleep by software. Clear by hardware
8:7	Reserved	Must be kept at reset value.
6	SRAM1P_WAKE	SRAM1 wakeup by software. Clear by hardware
5	SRAM1PSLEEP	SRAM1 go to sleep by software. Clear by hardware
4	Reserved	Must be kept at reset value.
3	WP_WAKE	Wi-Fi wakeup Only when WPEN bit is 1, set this bit by software will wakeup Wi-Fi. Or by clear WPEN bit (change from 1 to 0) can wakeup Wi-Fi. Clear by hardware.
2	WPSLEEP	Wi-Fi go to sleep by software only when WPEN bit is 1. Clear by hardware.
1	WPEN	Wi-Fi power enable (reset:1) (a global reset on Wi-Fi is needed after switching this bit) 1: Wi-Fi_OFF domain power off when Wi-Fi sleep, and power on when Wi-Fi wakeup.(when this bit is 1, Wi-Fi power on / off can be set by software or hardware) 0: Wi-Fi OFF domain power on. When the Wi-Fi OFF is power off, clear this bit will

wakeup Wi-Fi (Note that the IRC16M clock should be at work)

0 Reserved Must be kept at reset value.

5.4.4. Control and status register 1 (PMU_CS1)

Address offset: 0x0C

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

When PRIV in PMU_PRICFG register is 1, only privileged access is supported.

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	SRAM3PS_ACTIVE	SRAM3PS_SLEEP	Reserved	SRAM2PS_ACTIVE	SRAM2PS_SLEEP	Reserved	SRAM1PS_ACTIVE	SRAM1PS_SLEEP	Reserved	WPS_ACTIVE	WPS_SLEEP	Reserved			
r	r	r		r	r		r	r	r	r	r	r			

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	SRAM3PS_ACTIVE	SRAM3 is in active state. Read only
13	SRAM3PS_SLEEP	SRAM3 is in sleep state. Read only
12:11	Reserved	Must be kept at reset value.
10	SRAM2PS_ACTIVE	SRAM2 is in active state. Read only
9	SRAM2PS_SLEEP	SRAM2 is in sleep state. Read only
8:7	Reserved	Must be kept at reset value.
6	SRAM1PS_ACTIVE	SRAM1 is in active state. Read only
5	SRAM1PS_SLEEP	SRAM1 is in sleep state. Read only
4	Reserved	Must be kept at reset value.
3	WPS_ACTIVE	Wi-Fi is in active state. Read only
2	WPS_SLEEP	Wi-Fi is in sleep state. Read only
1:0	Reserved	Must be kept at reset value.

5.4.5. RF Control register (PMU_RFCTL)

Address offset: 0x20

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

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															

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	RFFC	Softw are set or clear (must > 2 IRC16M clock). Hardw are clear Softw are force close, close RF power, force to do the hardware RF sequence shutdown process.
1	RFFS	Softw are set or clear (must > 2 IRC16M clock). Hardw are clear Softw are force start, open RF power, force to do the hardware RF sequence opening process.
0	RFSWEN	1: RF sequence configured by softw are 0: RF sequence configured automatically by hardw are according to <u>Figure 4-6. RF sequence</u> (default)

5.4.6. Secure configuration register (PMU_SECCFG)

Address offset: 0x30

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

A non-secure write access is WI and generates an illegal access event. There are no read restrictions. When TZEN = 0, this register is RAZ/WI. When PRIV in PMU_PRICFG register is 1, only privileged access is supported.

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
rw															
Reserved				LPSSEC	DBPSEC	VDMSEC	LPMSEC	Reserved				WUP3SE	WUP2SE	WUPISE	WUPOSE
				rw	rw	rw	rw					rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.

16	RFSEC	RF security When set, the bits of PMU_RFCTL register is secure. A non-secure read / write access on secured bits is RAZ / WI.
15:12	Reserved	Must be kept at reset value.
11	LPSSEC	Wi-Fi_sleep and SRAM_sleep mode security When set, the bits of PMU_CTL1 register is secure. A non-secure read / write access on secured bits is RAZ / WI.
10	DBPSEC	Backup Domain write access security When set, BKPWEN bit in PMU_CTL0 register is secure. A non-secure read / write access on secured bits is RAZ / WI.
9	VDMSEC	Voltage detection and monitoring security When set, LVDEN, LVDT[2:0], VLVDEN bits in PMU_CTL0 register are secure. A non-secure read / write access on secured bits is RAZ / WI.
8	LPMSEC	Low-power mode security When set, LDOLP, STBMOD, WURST, STBRST bits in PMU_CTL0 register are secure. A non-secure read / write access on secured bits is RAZ / WI.
7:4	Reserved	Must be kept at reset value.
3	WUP3SEC	WKUP pin 3 security When set, WUPEN3 bit in PMU_CS0 register is secure. A non-secure read / write access on secured bits is RAZ / WI.
2	WUP2SEC	WKUP pin 2 security When set, WUPEN2 bit in PMU_CS0 register is secure. A non-secure read / write access on secured bits is RAZ / WI.
1	WUP1SEC	WKUP pin 1 security When set, WUPEN1 bit in PMU_CS0 register is secure. A non-secure read / write access on secured bits is RAZ / WI.
0	WUP0SEC	WKUP pin 0 security When set, WUPEN0 bit in PMU_CS0 register is secure. A non-secure read / write access on secured bits is RAZ / WI.

5.4.7. Privilege configuration register (PMU_PRICFG)

Address offset: 0x34

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

This register can be read by both privileged and unprivileged access. When TZEN = 1, and at least one bit in PMU_SECCFG register is set, this register can be read by both secure and non-secure access, and only secure write access is allowed.

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															PRIV

rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	PRIV	<p>Set and reset by software. This bit can be read by both privileged and unprivileged access. When set, it can only be cleared by a privileged access.</p> <p>0: All PMU registers can be read and written with privileged or unprivileged access.</p> <p>1: All PMU registers can be read and written only with privileged access. An unprivileged access to PMU registers is RAZ / WI. If the PMU is not secure, the PRIV bit can be written by a secure or non-secure privileged access. If TrustZone security is enabled (TZen = 1), if the PMU is secure, the PRIV bit can be written only by a secure privileged access: – A non-secure write access generates an illegal access event and write is ignored. – A secure unprivileged write access on PRIV bit is ignored.</p>

6. Reset and clock unit (RCU)

6.1. Reset control unit (RCTL)

6.1.1. Overview

GD32W51x Reset Control includes the control of three kinds of reset: power reset, system reset and backup domain reset. The power reset, known as a cold reset, resets the full system except the Backup domain. The system reset resets the processor core and peripheral IP components except for the SW-DP controller and the Backup domain. The backup domain reset resets the Backup domain. 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.

6.1.2. Function overview

Power reset

The Power reset is generated by either an external reset as Power On and Power Down reset (POR/PDR reset), Brownout reset (BOR reset) or by the internal reset generator when exiting Standby mode. The power reset sets all registers to their reset values except the Backup domain. The Power reset whose active signal is low, it will be de-asserted when the internal LDO voltage regulator is ready to provide 1.2V power.

System reset

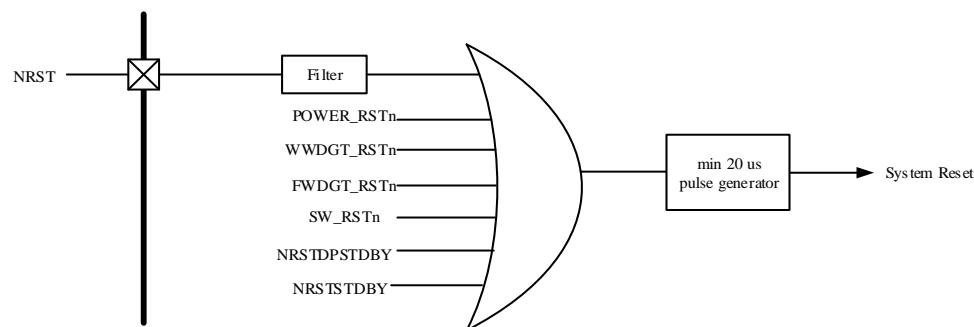
A system reset is generated by the following events:

- A power reset (POWER_RSTn)
- A 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)
- Reset generated when entering Standby mode when resetting NRSTSTDBY bit in EFUSE_USER_CTL register (NRSTSTDBY)
- Reset generated when entering Deep-sleep mode when resetting NRSTDPSLP bit in EFUSE_USER_CTL register (NRSTDPSLP)

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

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

Figure 6-1. The system reset circuit



Backup domain reset

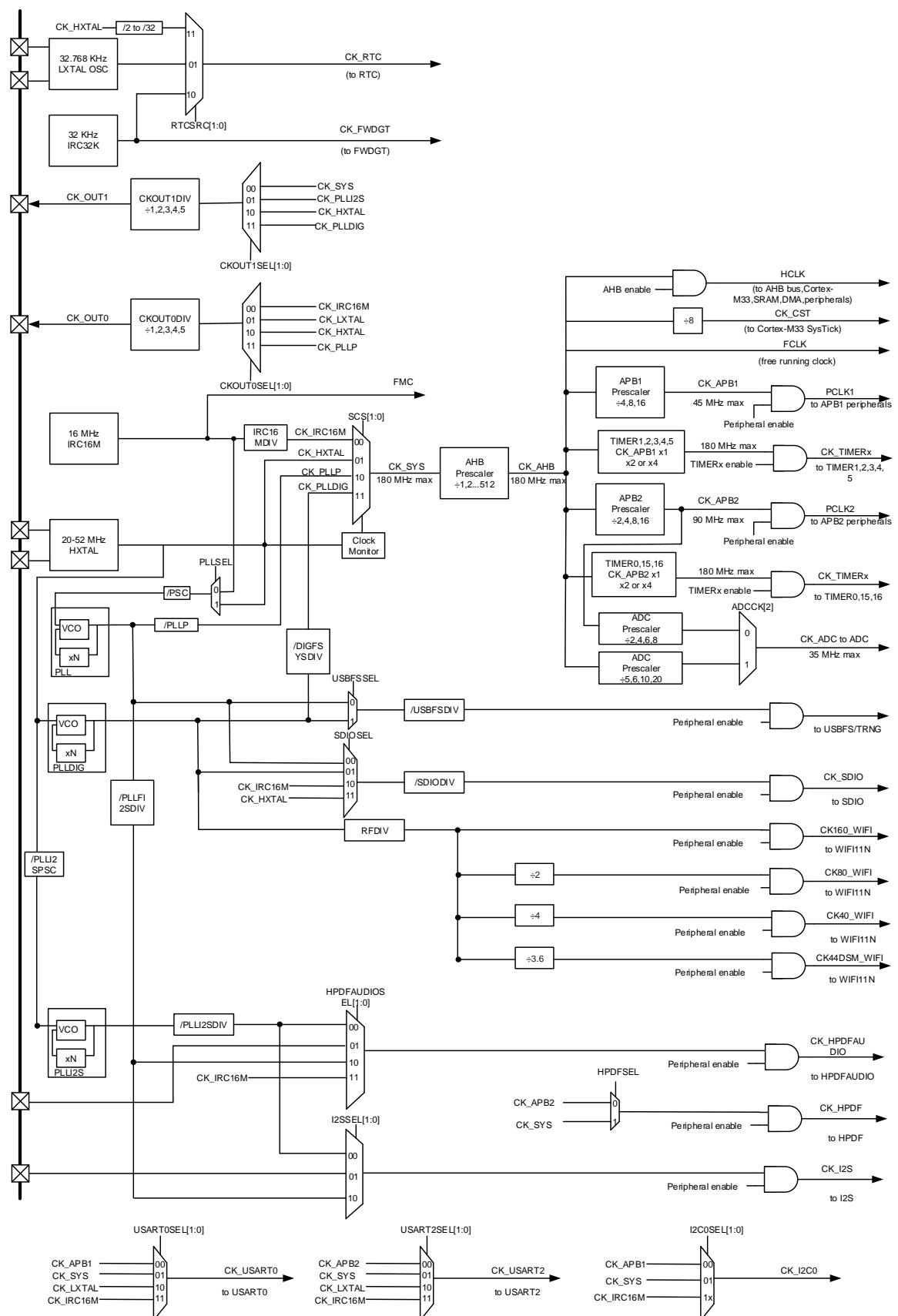
A backup domain reset is generated by setting the BKPRST bit in the Backup domain control register or Backup domain power on reset (V_{DD} or V_{BAT} power on, if both supplies have previously been powered off).

6.2. Clock control unit (CCTL)

6.2.1. Overview

The Clock Control unit provides a range of frequencies and clock functions. These include a Internal 16M RC oscillator (IRC16M), a High Speed crystal oscillator (HXTAL), a Low Speed Internal 32K RC oscillator (IRC32K), a Low Speed crystal oscillator (LXTAL), three Phase Lock Loop (PLL、PLLDIG、PLLI2S), two HXTAL clock monitor, 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 IRC16M, HXTAL or PLL (PLL or PLLDIG). The maximum operating frequency of the system clock (CK_SYS) can be up to 180MHz. The Free Watchdog Timer has independent clock source (IRC32K), and Real Time Clock (RTC) uses the IRC32K, LXTAL or HXTAL divided by RTCDIV (in RCU_CFG0 register) as its clock source.

Figure 6-2. Clock tree


The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB and the APB2/APB1 domains is 180MHz/90MHz/45 MHz. The Cortex-M33 System Timer (SysTick) external clock is clocked with the AHB clock (HCLK) divided by 8. The SysTick can work either with this clock or with the AHB clock (HCLK), configurable in the SysTick Control and Status Register.

The ADCs are clocked by the clock of APB2 divided by 2, 4, 6, 8 or by the clock of AHB divided by 5, 6, 10, 20, which defined by ADCCK in ADC_CCTL register.

The TIMERS are clocked by the clock divided from CK_AHB. The frequency of TIMERS clock is equal to CK_APBx, twice the CK_APBx or four times the CK_APBx. Please refer to TIMERSEL bit in RCU_CFG1 for detail.

The USBFS /TRNG/ clocks are selected from the clock of PLL or PLLDIG.

The SDIO is clocked by PLL or PLLDIG or IRC16M clock or HXTAL clock, which selected by SDIOSEL bits in RCU_ADDCTL register.

The I2S is clocked by the clock of PLLI2SR or External PIN I2S_CKIN which defined by I2SSEL bit in RCU_ADDCTL register.

The RTC is clocked by LXTAL clock or IRC32K clock or HXTAL clock divided by 2 to 32 (defined by RTCDIV bits in RCU_CFG0) which select by RTCSR bit in Backup Domain Control Register (RCU_BDCTL). After the RTC select HXTAL clock divided by 2 to 31 (defined by RTCDIV bits in RCU_CFG0), the clock disappeared when the 1.2V core domain power off. After the RTC select IRC32K, the clock disappeared when V_{DD} power off. When the RTC select LXTAL, the clock disappeared when V_{DD} and V_{BAT} power off.

The FWDGT is clocked by IRC32K clock, which is forced on when FWDGT started.

The USART0/2 is clocked by IRC16M clock or LXTAL clock or System clock or APB2 clock, which selected by USART0SEL bits in RCU_CFG1 register.

The I2C0 is clocked by IRC16M clock or System clock or APB1 clock, which selected by I2C0SEL bits in RCU_CFG1 register.

The HPDF is clocked by PCLK2 clock or System clock, which selected by HPDFSEL bit in RCU_ADDCTL register.

The HPDF_AUDIO is clocked by PLLI2S or External I2S_CKIN PIN or PLL or IRC16M which defined by HPDFADUDIOSEL bit in RCU_ADDCTL register.

6.2.2. Characteristics

- 20 to 52 MHz High Speed crystal oscillator (HXTAL)
- Internal 16 MHz RC oscillator (IRC16M)
- 32,768 Hz Low Speed crystal oscillator (LXTAL)
- Internal 32KHz RC oscillator (IRC32K)
- PLL clock source can be HXTAL or IRC16M

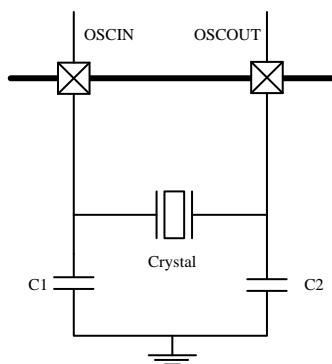
- PLLDIG clock source can be HXTAL
- PLLI2S clock source can be HXTAL
- HXTAL clock monitor

6.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed external crystal oscillator (HXTAL), which has a frequency from 20 to 52 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 6-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN/HXTALP U/HXTALREADY 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. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Internal 16M RC oscillators (IRC16M)

The internal 16M RC oscillator, IRC16M, has a fixed frequency of 16 MHz and is the default clock source selection for the CPU when the device is powered up. The IRC16M oscillator provides a lower cost type clock source as no external components are required. The IRC16M RC oscillator can be switched on or off using the IRC16MEN bit in the Control Register RCU_CTL. The IRC16MSTB flag in the Control Register RCU_CTL is used to indicate if the

internal 16M RC oscillator is stable. The start-up time of the IRC16M oscillator is shorter than the HXTAL crystal oscillator. An interrupt can be generated if the related interrupt enable bit, IRC16MSTBIE, in the Clock Interrupt Register, RCU_INT, is set when the IRC16M becomes stable. The IRC16M clock can also be used as the system clock source or the PLL input clock.

The frequency accuracy of the IRC16M 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 IRC16M clock to be the system clock when the system initially wakes-up.

If USART0、USRAT2 and I2C0 select IRC16M as function clock, the IRC16M will be force on according to USART0、USRAT2 and I2C0 function.

Phase locked loop (PLL)

There are three internal Phase Locked Loop, the PLL, PLLI2S and PLLDIG. The PLLP and PLLDIG could be used to generator system clock (no more than 180MHz) and their division-clock which used to USBFS/TRNG/SDIO/I2S/HPDF. The PLLI2S is used to generator the clock to I2S/HPDF.

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

The PLLI2S can be switched on or off by using the PLLI2SEN bit in the RCU_CTL Register. The PLLI2SSSTB flag in the RCU_CTL Register will indicate if the PLLI2S clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLI2SSSTBIE, in the RCU_INT Register, is set as the PLLI2S becomes stable.

The PLLDIG can be switched on or off by using the PLLDIGEN/PLLDIGPU bit in the RCU_CTL Register. The PLLDIGSTB flag in the RCU_CTL Register will indicate if the PLLDIG clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLDIGSTBIE, in the RCU_INT Register, is set as the PLLDIG becomes stable.

The three PLLs are closed by hardware when entering the DeepSleep/Standy mode or HXTAL monitor fail when HXTAL used as the source clock of the PLLs.

Low speed crystal oscillator (LXTAL)

The low speed external crystal or ceramic resonator oscillator, which has a frequency of 32,768 Hz, produces a low power but highly accurate clock source for the Real Time Clock circuit. The LXTAL oscillator can be switched on or off using the LXTALEN bit in the Backup Domain Control Register (RCU_BDCTL). The LXTALSTB flag in the Backup Domain Control Register (RCU_BDCTL) will indicate if the LXTAL clock is stable. An interrupt can be

generated if the related interrupt enable bit, LXTALSTBIE, in the Interrupt Register RCU_INT is set when the LXTAL becomes stable.

Select external clock bypass mode by setting the LXTALBPS and LXTALEN bits in the Backup Domain Control Register (RCU_BDCTL). The CK_LXTAL is equal to the external clock which drives the OSC32IN pin.

Internal 32K RC oscillator (IRC32K)

The internal RC oscillator has a frequency of about 32 kHz 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 Clock 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 IRC16M and can be switched to HXTAL or CK_PLLP or CK_PLLDIG by changing the System Clock Switch bits, SCS, in the Clock 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 directly or indirectly (by PLL or PLLDIG) used as the CK_SYS, it is not possible to stop it.

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, the HXTAL will be automatically disabled. The HXTAL Clock Stuck interrupt Flag, CKMIF, in the Clock Interrupt Register, RCU_INT, will be set and the HXTAL failure event will be generated. 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 and CK_PLLP used as system clock, the HXTAL failure will force the CK_SYS source to IRC16M and the PLL will be disabled automatically. If the HXTAL is selected as the clock source of any PLLs, the HXTAL failure will force the PLL closed automatically.

Clock output capability

The clock output capability is ranging from 32kHz to 180MHz. There are several clock signals can be selected via the CK_OUT0 clock source selection bits, CKOUT0SEL, in the Clock 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. The

CK_OUT1 is selected by CKOUT1SEL, in the Clock Configuration Register 0 (RCU_CFG0).

Table 6-1. Clock output 0 source select

Clock Source 0 Selection bits	Clock Source
00	CK_IRC16M
01	CK_LXTAL
10	CK_HXTAL
11	CK_PLLP

Table 6-2. Clock output 1 source select

Clock Source 1 Selection bits	Clock Source
00	CK_SYS
01	CK_PLLI2SR
10	CK_HXTAL
11	CK_PLLDIG

The CK_OUT0 frequency can be reduced by a configurable binary divider, controlled by the CKOUT0DIV bits, in the Clock Configuration Register (RCU_CFG0).

The CK_OUT1 frequency can be reduced by a configurable binary divider, controlled by the CKOUT1DIV bits, in the Clock Configuration Register (RCU_CFG0).

Voltage control

The 1.2V domain voltage in Deep-sleep mode can be controlled by DSLPVS[2:0] bit in the Deep-sleep mode voltage register (RCU_DSV). 1.2V domain voltage selected in deep-sleep mode

Table 6-3. 1.2V domain voltage selected in deep-sleep mode

DSLPVS[1:0]	Deep-sleep mode voltage(V)
00	1.1
01	1.0
10	0.9
11	0.8

The RCU_DSV register are protected by Voltage Key register (RCU_VKEY). Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_DSV register can be written.

6.3. RCU security protection

When the TrustZone security is enable, the RCU_SECP_CFG is able to RCU secure registers from being modified by non-secure accesses. When the Trustzone security is disabled, all registers are non-secure. The RCU_SECP_CFG secure register and security status registers are RAZ/WI. The TrustZone security is activated by the TZEN option bit in the FMC_OBR register.

By configuring the RCU_SECP_CFG register, you can prohibit non-secure access to read or modify the configuration and status of HXTAL, CKMEN, IRC16M, IRC32K, LXTAL, PLL, PLLDIG, PLLI2S, AHB pre-scaler related registers, as well as the selection of the system clock source, CKOUT clock output and reset flag RMVF can also be configured through the RCU_SECP_CFG register. The summary of the RCU secured bits in RCU_SECP_CFG register is show as the [Table 6-4. RCU secure protection configuration summary](#).

For a secure peripheral, its related clock, reset, clock source selection and clock enable during low-power modes control bits are also secure in the RCU_AHbxEN, RCU_APBxEN, RCU_AHbxRST, RCU_APBxRST, RCU_AHbxSPEN, RCU_APBxSPEN registers.

Table 6-4. RCU secure protection configuration summary

Configuration bit in RCU_SECP_CFG	Corresponding register	Secured bit
IRC16MSECP=1	RCU_CTL	IRC16MEN, IRC16MSTB, IRC16MA DJ, IRC16MCALIB,
	RCU_INT	IRC16MSTBIE, IRC16MSTBIF, IRC16MSTBIC, IRC16MSTBIF
	RCU_CFG1	IRC16MDIV
HXTALSECP=1	RCU_CTL	HXTALEN, HXTALSTV, HXTALBPS, CKMEN, RFCKME N, HXTALPU, HXTALENI2S, HSTALENPLL, HA TALREA DY
	RCU_INT	CKMIC, HXTALSTBIC, HXTALSTBIE, CKMIF, HXTALSTBIF
IRC32KSECP=1	RCU_PLLSSCTL	IRC32KEN, IRC32KSTB
	RCU_INT	IRC32KSTBIC, IRC32KSTBIE, IRC32KSTBIF
LXTALSECP=1	RCU_BDCTL	LXTALEN, LXTALSTB, LXTALBPS, LXTALDRI
SYSCLKSECP=1	RCU_CFG0	SCS[1:0], SCSS[1:0], CKOUT0SEL[1:0], CKOUT0DIV[2:0], CKOUT1DIV[2:0], CKOUT1SEL[1:0]
PRESCECP=1	RCU_CFG0	AHBPSC[3:0], APB1PSC[2:0], APB2PSC [2:0], TIMERSEL
PLLSECP=1	RCU_PLL	PLLSRC[0], PLLPSC[6:0], PLLN[8:0], PLLEN, PLLP[1:0],
	RCU_CTL	PLLSTB, PLLEN
	RCU_PLLSSCTL	SSCGON, SS_TYPE, MODSTEP, MODCNT
	RCU_INT	PLLSTBIC, PLLSTBIE, PLLSTBIF
PLLDIGSECP=1	RCU_CTL	PLLDIGFSYSDIV[5:0], PLLDIGSEL[1:0]
	RCU_CTL	PLLDIGSTB, PLLDIGEN, PLLDIGPU

Configuration bit in RCU_SECP_CFG	Corresponding register	Secured bit
	RCU_INT	PLLDIGSTBIC, PLLDIGSTBIE, PLLDIGSTBIF
RMVFSECP=1	RCU_RSTSCK	RSTFC
BKPSECP=1	RCU_BDCTL	BKPRST
PLL2SSSEC=1	RCU_PLLCFG2	PLL2SN[6:0], PLL2SPSC[2:0], PLL2SDIV[5:0]
	RCU_CTL	PLL2SSTB, PLL2SEN
	RCU_ADDCTL	PLLFI2SDIV
	RCU_INT	PLL2SSTBIC, PLL2SSTBIE, PLL2SSTBIF

When a certain bit register in RCU_SECP_CFG is set, some RCU registers will have security attributes, and the access to the secure and non-secure bits in the register will follow the rules in [Table 6-5. RCU secure-bit or nonsecure-bit access rules](#).

Table 6-5. RCU secure-bit or nonsecure-bit access rules

Access mode	Read		Write	
	Secure access	Nonsecure access	Secure access	Nonsecure access
Secure-bit	allowed	RAZ(no illegal access event)	allowed	WI(no illegal access event)
Nonsecure-bit	allowed	allowed	allowed	allowed
RCU_SECP_CFG	allowed	RAZ(generates an illegal access event) ⁽¹⁾	allowed	WI(generates an illegal access event) ⁽¹⁾

NOTE: (1) An illegal access interrupt is generated if the RCU illegal access interrupt is enabled in the TZIAC_INTEN2 register.

6.4. RCU privilege protection

The rules for accessing RCU registers in privileged and non-privileged protection modes are shown in [Table 6-6. RCU register privileg and unprivileg access rules](#).

Table 6-6. RCU register privileg and unprivileg access rules

Access mode	Read		Write	
	Priviledg access	Unprivileg access	Priviledg access	Unprivileg access
RCUPRIP = 0	allowed	allow ed(expect RCUPRIP bit in RCU_CTL)	allowed	allow ed(expect RCUPRIP bit in RCU_CTL)
RCUPRIP = 1	allow ed(except RCU_AHBxSECP_STAT, RCU_APBx_SECP_STAT, RCU_	RAZ	allow ed	WI

Access mode	Read		Write	
	Priviledg access	Unprivileg access	Priviledg access	Unprivileg access
	SECP_STAT,RCUP (RIP in RCU_CTL)			

6.5. Register definition

RCU secure access base address: 0x5002 3800
 RCU non-secure access base address: 0x4002 3800

6.5.1. Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0004 xx83 where x is undefined.

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
HXTALR	HXTALE	HXTALE	HXTALP	PLLI2SST	PLLI2SE	PLLSTB	PLLEN	PLLDIGS	RFCKME	PLLDIGE	PLLDIGP	CKMEN	HXTALB	HXTALST	HXTALE
EADY	NPLL	NI2S	U	B	N			TB	N	N	U	PS	B		N
rw	rw	r	rw	r	rw	r	rw	r	rw	rw	rw	rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC16MCALIB[7:0]								IRC16MADJ[4:0]				RCUPRP	IRC16MS	IRC16ME	
r								rw				rw	r	rw	

Bits	Fields	Descriptions
31	HXTALREADY	High Speed crystal oscillator ready set by software, which can not be written when HXTALEN, HXTALEN12S, HXTALENPLL or PLLDIGEN is enable. 0: HXTAL is not ready set by software 1: HXTAL is ready set by software
30	HXTALENPLL	High Speed crystal oscillator enable for system CK_PLLP, which can be written when PLL is off Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: High speed crystal oscillator for CK_PLLP disable 1: High speed crystal oscillator for CK_PLLP enable
29	HXTALEN12S	High Speed crystal oscillator enable for PLLI2S, which can be written when PLLI2S is off Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: High speed crystal oscillator for PLLI2S disable 1: High speed crystal oscillator for PLLI2S enable
28	HXTALPU	High Speed crystal oscillator (HXTAL) Power Up, which can be written when HXTALEN, HXTALEN12S and HXTALENPLL or PLLDIGEN are on. Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode.

		0: High speed crystal oscillator Power down 1: High speed crystal oscillator Power up
27	PLL12SSTB	<p>PLL12S clock stabilization flag</p> <p>Set by hardware to indicate if the PLL12S output clock is stable and ready for use.</p> <p>0: PLL12S is not stable 1: PLL12S is stable</p>
26	PLL12SEN	<p>PLL12S enable</p> <p>Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode.</p> <p>0: PLL12S disable 1: PLL12S enable</p>
25	PLLSTB	<p>PLL clock stabilization flag</p> <p>Set by hardware to indicate if the PLL output clock is stable and ready for use.</p> <p>0: PLL is not stable 1: PLL is stable</p>
24	PLLEN	<p>PLL enable</p> <p>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.</p> <p>0: PLL disable 1: PLL enable</p>
23	PLLDIGSTB	<p>PLLDIG Clock Stabilization Flag</p> <p>Set by hardware to indicate if the PLLDIG output clock is stable and ready for use.</p> <p>0: PLLDIG is not stable 1: PLLDIG is stable</p>
22	RFCKMEN	<p>HXTAL Clock Monitor Enable, Check RF XTAL</p> <p>0: Disable the High speed crystal oscillator (HXTAL) clock monitor 1: Enable the High speed crystal oscillator (HXTAL) clock monitor</p>
21	PLLDIGEN	<p>PLLDIG enable, This bit cannot be reset if the PLLDIG clock is used as the system clock</p> <p>Set and reset by software.</p> <p>0: PLLDIG disable 1: PLLDIG enable</p>
20	PLLDIGPU	<p>PLLDIG power up, This bit cannot be reset if the PLLDIG clock is used as the system clock.</p> <p>Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode.</p> <p>0: PLLDIG power down. 1: PLLDIG power up.</p>
19	CKMEN	HXTAL clock monitor enable

0: Disable the High speed 20 ~ 52 MHz crystal oscillator (HXTAL) clock monitor
 1: Enable the High speed 20 ~ 52 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 IRC16M 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 IRC16M internal RC oscillator regardless of the control bit, IRC16MEN, state.

18	HXTALBPS	High speed crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN and HXTALPU both are 0. 0: Enable the HXTAL Bypass mode in which the HXTAL output clock is equal to the input clock. 1: Disable the HXTAL Bypass mode.
17	HXTALSTB	High speed 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
16	HXTALEN	High Speed crystal oscillator (HXTAL) 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 when PLL clock is selected to the system clock. Reset by hardware when entering Standby mode. If enable PLLDIG or RFPLL, this bit need be set to 1. 0: High speed crystal oscillator disabled 1: High speed crystal oscillator enabled
15:8	IRC16MCALIB[7:0]	Internal 16MHz RC Oscillator calibration value register These bits are load automatically at power on.
7:3	IRC16MA DJ[4:0]	Internal 16MHz RC Oscillator clock trim adjust value These bits are set by software. The trimming value is these bits (IRC16MA DJ) added to the IRC16MCALIB[7:0] bits. The trimming value should trim the IRC16M to 16 MHz \pm 1%.
2	RCUPRIP	RCU privilege protection Set and reset by software. This bit can be read by both privileged or unprivileged access. When set, it can only be cleared by a privileged access. 0: RCU registers can be accessed by a privileged or non-privileged access. 1: RCU registers can be accessed only by a privileged access except RCU_AHBxSECSTAT, RCU_APBxSECSTAT and RCU_SECSTAT. An unprivileged access to RCU registers is RAZ/WI. If TrustZone security is enabled (TZEN = 1), when the RCU is not secure, the RCUPRIV bit can be written by a secure or non-secure privileged access. If the RCU is secure, the RCUPRIV bit can be written only by a secure privileged access.

		nonsecure write is ignored.
		A secure unprivileged write access on RCUPRIV bit is ignored.
1	IRC16MSTB	IRC16M Internal 16MHz RC Oscillator stabilization flag Set by hardware to indicate if the IRC16M oscillator is stable and ready for use. 0: IRC16M oscillator is not stable 1: IRC16M oscillator is stable
0	IRC16MEN	Internal 16MHz RC oscillator enable Set and reset by software. This bit cannot be reset if the IRC16M 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 CKMEN is set. 0: Internal 16 MHz RC oscillator disabled 1: Internal 16 MHz RC oscillator enabled

6.5.2. PLL register (RCU_PLL)

Address offset: 0x04

Reset value: 0x0000 3010

To configure the PLL clock, refer to the following formula:

$$CK_{PLLVCOSRC} = CK_{PLLSRC} / PLLPSC$$

$$CK_{PLLVCO} = CK_{PLLVCOSRC} \times PLLN$$

$$CK_{PLLP} = CK_{PLLVCO} / PLLP$$

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	PLLP[1:0]	The PLLP output frequency division factor from PLL VCO clock Set and reset by software when the PLL is disable. These bits used to generate PLLP output clock (CK_PLLP) from PLL VCO clock (CK_PLLVCO). The CK_PLLP is used to system clock (no more than 180MHz). The CK_PLLVCO is described in PLLN bits in RCU_PLL register. 00 : CK_PLLP = CK_PLLVCO / 2 01 : CK_PLLP = CK_PLLVCO / 4 10 : CK_PLLP = CK_PLLVCO / 6 11 : CK_PLLP = CK_PLLVCO / 8

15	PLLSEL	PLL clock source selection Set and reset by softw are to control the PLL clock source. 0: IRC16M clock selected as source clock of PLL 1: HXTAL clock selected as source clock of PLL
14:6	PLLN[8:0]	The PLL VCO clock multiplication factor Set and reset by softw are (only use w ord/half-word w rite) when the PLL is disable. These bits used to generator PLL VCO clock (CK_PLLVCO) from PLL VCO source clock (CK_PLLVCOSRC). The CK_PLLVCOSRC is described in PLLPSC bits in RCU_PLL register. Note: The frequency of CK_PLLVCO is between 64MHz to 400MHz The value of PLLN must : 64≤PLLN≤511 (when SSCGON=0 in RCU_PLLSCTL) 71≤PLLN≤504 (when SSCGON=1/SS_TYPE=0 in RCU_PLLSCTL) 75≤PLLN≤508 (when SSCGON=1/SS_TYPE=1 in RCU_PLLSCTL) 00000000: Reserved 00000001: Reserved ... 00011111: Reserved 00100000: CK_PLLVCO = CK_PLLVCOSRC x 64. 00100001: CK_PLLVCO = CK_PLLVCOSRC x 65. ... 11111110: CK_PLLVCO = CK_PLLVCOSRC x 510. 11111111: CK_PLLVCO = CK_PLLVCOSRC x 511.
5:0	PLLPSC[5:0]	The PLL VCO source clock prescaler Set and reset by softw are when the PLL is disable. These bits used to generate the clock of PLL VCO source clock (CK_PLLVCOSRC). The PLL VCO source clock is between 1M to 2MHz. 00000: Reserved. 00001: Reserved 00010: CK_PLLSRC / 2 00011: CK_PLLSRC / 3 ... 11111: CK_PLLSRC / 63

6.5.3. Clock configuration register 0 (RCU_CFG0)

Address offset: 0x08

Reset value: 0x0000 9400

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
CKOUT1SEL[1:0]	CKOUT1DIV[2:0]	CKOUT0DIV[2:0]	Reserved	CKOUT0SEL[1:0]	RTCDIV[4:0]										

Bits	Fields	Descriptions
31:30	CKOUT1SEL[1:0]	<p>CKOUT1 clock source selection</p> <p>Set and reset by software.</p> <p>00: System clock selected</p> <p>01: CK_PLLI2S clock selected (SPI1EN or HPDFEN must be 1, to generate CK_PLLI2S)</p> <p>10: High Speed crystal oscillator clock (HXTAL) selected</p> <p>11: CK_PLLDIG clock selected</p>
29:27	CKOUT1DIV[2:0]	<p>The CK_OUT1 divider which the CK_OUT1 frequency can be reduced see bits 31:30 of RCU_CFG0 for CK_OUT1</p> <p>0xx: The CK_OUT1 is divided by 1</p> <p>100: The CK_OUT1 is divided by 2</p> <p>101: The CK_OUT1 is divided by 3</p> <p>110: The CK_OUT1 is divided by 4</p> <p>111: The CK_OUT1 is divided by 5</p>
26:24	CKOUT0DIV[2:0]	<p>The CK_OUT0 divider which the CK_OUT0 frequency can be reduced see bits 22:21 of RCU_CFG0 for CK_OUT0</p> <p>0xx: The CK_OUT0 is divided by 1</p> <p>100: The CK_OUT0 is divided by 2</p> <p>101: The CK_OUT0 is divided by 3</p> <p>110: The CK_OUT0 is divided by 4</p> <p>111: The CK_OUT0 is divided by 5</p>
23	Reserved	Must be kept at reset value.
22:21	CKOUT0SEL[1:0]	<p>CKOUT0 clock source selection</p> <p>Set and reset by software.</p> <p>00: Internal 16M RC Oscillator clock selected</p> <p>01: Low Speed crystal oscillator clock (LXTAL) selected</p> <p>10: High Speed crystal oscillator clock (HXTAL) selected</p> <p>11: CK_PLLP clock selected</p>
20:16	RTCDIV[4:0]	<p>RTC clock divider factor</p> <p>Set and reset by software. These bits is used to generator clock for RTC (no more than 1MHz) from HXTAL clock.</p> <p>00000: no clock for RTC</p> <p>00001: CK_HXTAL / 2</p> <p>00010: CK_HXTAL / 3</p> <p>00011: CK_HXTAL / 4</p>

		...
		11111: CK_HXTAL / 32
15:13	APB2PSC[2:0]	<p>APB2 prescaler selection</p> <p>Set and reset by softw are to control the APB2 clock division ratio.</p> <p>0xx: Reserved</p> <p>100: (CK_AHB / 2) selected</p> <p>101: (CK_AHB / 4) selected</p> <p>110: (CK_AHB / 8) selected</p> <p>111: (CK_AHB / 16) selected</p>
12:10	APB1PSC[2:0]	<p>APB1 prescaler selection</p> <p>Set and reset by softw are to control the APB1 clock division ratio.</p> <p>0xx: Reserved</p> <p>100: Reserved</p> <p>101: (CK_AHB / 4) selected</p> <p>110: (CK_AHB / 8) selected</p> <p>111: (CK_AHB / 16) selected</p>
9:8	Reserved	Must be kept at reset value.
7:4	AHBPSC[3:0]	<p>AHB prescaler selection</p> <p>Set and reset by softw are to control the AHB clock division ratio</p> <p>0xxx: CK_SYS selected</p> <p>1000: (CK_SYS / 2) selected</p> <p>1001: (CK_SYS / 4) selected</p> <p>1010: (CK_SYS / 8) selected</p> <p>1011: (CK_SYS / 16) selected</p> <p>1100: (CK_SYS / 64) selected</p> <p>1101: (CK_SYS / 128) selected</p> <p>1110: (CK_SYS / 256) selected</p> <p>1111: (CK_SYS / 512) selected</p>
3:2	SCSS[1:0]	<p>System clock sw itch status</p> <p>Set and reset by hardw are to indicate the clock source of system clock.</p> <p>00: select CK_IRC16M as the CK_SYS source</p> <p>01: select CK_HXTAL as the CK_SYS source</p> <p>10: select CK_PLLP as the CK_SYS source</p> <p>11: select CK_PLLDIG as the CK_SYS source</p>
1:0	SCS[1:0]	<p>System clock sw itch</p> <p>Set by softw are to select the CK_SYS source. Because the change of CK_SYS has inherent latency, softw are should read SCSS to confirm whether the switching is complete or not. The switch will be forced to IRC16M when leaving Deep-sleep and Standby mode or HXTAL failure is detected by HXTAL clock monitor when HXTAL is selected directly or indirectly as the clock source of CK_SYS</p> <p>00: select CK_IRC16M as the CK_SYS source</p>

01: select CK_HXTAL as the CK_SYS source

10: select CK_PLLP as the CK_SYS source

11: select CK_PLLDIG as the CK_SYS source

6.5.4. Clock interrupt register (RCU_INT)

Address offset: 0x0C

Reset value: 0x0000 0000

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	PLLDIGS	PLL2SS	PLLSTBI	HXTALST	IRC16MS	LXTALST	IRC32KS
									TBIC	TBIC	C	BIC	TBIC	BIC	TBIC	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	PLLDIGS	PLL2SS	PLLSTBI	HXTALST	IRC16MS	LXTALST	IRC32KS	CKMIF	PLLDIGS	PLL2SS	PLLSTBI	HXTALST	IRC16MS	LXTALST	IRC32KS	
	TBIE	BIE	E	BIE	TBIE	BIE	TBIE		TBIF	TBIF	F	BIF	TBIF	BIF	TBIF	
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	

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	PLLDIGSTBIC	PLLDIG stabilization interrupt clear Write 1 by software to reset the PLLDIGSTBIF flag. 0: Not reset PLLDIGSTBIF flag 1: Reset PLLDIGSTBIF flag
21	PLL2SSSTBIC	PLL2S stabilization interrupt clear Write 1 by software to reset the PLL2SSSTBIF flag. 0: Not reset PLL2SSSTBIF flag 1: Reset PLL2SSSTBIF flag
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	IRC16MSTBIC	IRC16M stabilization interrupt clear Write 1 by software to reset the IRC16MSTBIF flag. 0: Not reset IRC16MSTBIF flag 1: Reset IRC16MSTBIF flag
17	LXTALSTBIC	LXTAL stabilization interrupt clear Write 1 by software to reset the LXTALSTBIF flag. 0: Not reset LXTALSTBIF flag 1: Reset LXTALSTBIF flag
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	PLLDIGSTBIE	PLLDIG stabilization interrupt enable Set and reset by software to enable/disable the PLLDIG stabilization interrupt. 0: Disable the PLLDIG stabilization interrupt 1: Enable the PLLDIG stabilization interrupt
13	PLLl2SSTBIE	PLLl2S Stabilization interrupt enable Set and reset by software to enable/disable the PLLl2S stabilization interrupt. 0: Disable the PLLl2S stabilization interrupt 1: Enable the PLLl2S stabilization interrupt
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	IRC16MSTBIE	IRC16M Stabilization interrupt enable Set and reset by software to enable/disable the IRC16M stabilization interrupt 0: Disable the IRC16M stabilization interrupt 1: Enable the IRC16M stabilization interrupt
9	LXTALSTBIE	LXTAL Stabilization interrupt enable LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt

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 when setting the CKMIC bit by software. 0: Clock operating normally 1: HXTAL clock stuck
6	PLLDIGSTBIF	PLLDIG stabilization interrupt flag Set by hardware when the PLLDIG is stable and the PLLDIGSTBIE bit is set. Reset when setting the PLLDIGSTBIC bit by software. 0: No PLLDIG stabilization interrupt generated 1: PLLDIG stabilization interrupt generated
5	PLLl2SSTBIF	PLLl2S stabilization interrupt flag Set by hardware when the PLLl2S is stable and the PLLl2SSTBIE bit is set. Reset when setting the PLLl2SSTBIC bit by software. 0: No PLLl2S stabilization interrupt generated 1: PLLl2S stabilization interrupt generated
4	PLLSTBIF	PLL stabilization interrupt flag Set by hardware when the PLL is stable and the PLLSTBIE bit is set. Reset when setting the PLLSTBIC bit by software. 0: No PLL stabilization interrupt generated 1: PLL stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the High speed 20 ~ 52 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset when setting the HXTALSTBIC bit by software. 0: No HXTAL stabilization interrupt generated 1: HXTAL stabilization interrupt generated
2	IRC16MSTBIF	IRC16M stabilization interrupt flag Set by hardware when the Internal 16 MHz RC oscillator clock is stable and the IRC16MSTBIE bit is set. Reset when setting the IRC16MSTBIC bit by software. 0: No IRC16M stabilization interrupt generated 1: IRC16M stabilization interrupt generated
1	LXTALSTBIF	LXTAL stabilization interrupt flag Set by hardware when the Low speed 32,768 Hz crystal oscillator clock is stable and the LXTALSTBIE bit is set. Reset when setting the LXTALSTBIC bit by software.

		0: No LXTAL stabilization interrupt generated 1: LXTAL stabilization interrupt generated
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 when setting the IRC32KSTBIC bit by software. 0: No IRC32K stabilization clock ready interrupt generated 1: IRC32K stabilization interrupt generated

6.5.5. AHB1 reset register (RCU_AHB1RST)

Address offset: 0x10

Reset value: 0x0000 0000

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	USBFSRST								DMA1RS	DMA0RS					
	ST								T	T					
rw									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WIFIRST	CRCRST			Reserved		TSIRST	TZGPCRST			Reserved		PCRST	PBRST	PARST
	rw	rw					rw	rw					rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	USBFSRST	USBFS reset This bit is set and reset by software. 0: No reset 1: Reset the USBFS
28:23	Reserved	Must be kept at reset value
22	DMA1RST	DMA1 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA1
21	DMA0RST	DMA0 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA0
20:14	Reserved	Must be kept at reset value

13	WIFIRST	Wi-Fi reset This bit is set and reset by software. 0: No reset 1: Reset the Wi-Fi
12	CRCRST	CRC reset This bit is set and reset by software. 0: No reset 1: Reset the CRC
11:9	Reserved	Must be kept at reset value
8	TSIRST	TSI reset This bit is set and reset by software. 0: No reset 1: Reset the TSI
7	TZGPCRST	TZGPC reset This bit is set and reset by software. 0: No reset 1: Reset the TZGPC
6:3	Reserved	Must be kept at reset value
2	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port C
1	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port B
0	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port A

6.5.6. AHB2 reset register (RCU_AHB2RST)

Address offset: 0x14

Reset value: 0x0000 0000

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TRNGRS T	HAURST	CAURST	PKCAUR ST		Reserved	DCIRST

rw rw rw rw rw rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	TRNGRST	TRNG reset This bit is set and reset by software. 0: No reset 1: Reset the TRNG
5	HAURST	HAU reset This bit is set and reset by software. 0: No reset 1: Reset the HAU
4	CAURST	CAU reset This bit is set and reset by software. 0: No reset 1: Reset the CAU
3	PKCAURST	PKCAU reset This bit is set and reset by software. 0: No reset 1: Reset the PKCAU
2:1	Reserved	Must be kept at reset value
0	DCIRST	DCI reset This bit is set and reset by software. 0: No reset 1: Reset the DCI

6.5.7. AHB3 reset register (RCU_AHB3RST)

Address offset: 0x18

Reset value: 0x0000 0000

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

Reserved

QSPIRST SQPIRST

rw rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	QSPIRST	<p>QSPI reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the QSPI</p>
0	SQPIRST	<p>SQPI reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the SQPI</p>

6.5.8. APB1 reset register (RCU_APB1RST)

Address offset: 0x20

Reset value: 0x0000 0000

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		PMURST		Reserved		I2C1RST	I2C0RST	Reserved	USART0 RST	USART1 RST	Reserved				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1RST	Reserved	WWWDGT RST		Reserved				TIMER5R ST	TIMER4R ST	TIMER3R ST	TIMER2R ST	TIMER1R ST		

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	PMURST	<p>PMU reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the PMU</p>
27:23	Reserved	Must be kept at reset value
22	I2C1RST	<p>I2C1 reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the I2C1</p>

21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C0
20:19	Reserved	Must be kept at reset value
18	USART0RST	USART0 reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset the USART1
16:15	Reserved	Must be kept at reset value.
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI1
13:12	Reserved	Must be kept at reset value.
11	WWDGTRST	WWDGTRST reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGTRST
10:5	Reserved	Must be kept at reset value.
4	TIMER5RST	TIMER5 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER5
3	TIMER4RST	TIMER4 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER4
2	TIMER3RST	TIMER3 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER3
1	TIMER2RST	TIMER2 reset This bit is set and reset by software.

0: No reset

1: Reset the TIMER2

0 **TIMER1 RST** **TIMER1 reset**

This bit is set and reset by software.

0: No reset

1: Reset the TIMER1

6.5.9. APB2 reset register (RCU_APB2RST)

Address offset: 0x24

Reset value: 0x0000 0000

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
RFRST	HPDFRS T												TIMER16 RST	TIMER15 RST	Reserved
rw	rw												rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYSCFG RST	Reserved	SPI0RST	SDIORST	Reserved	ADC0RS T		Reserved		USART2 RST		Reserved		TIMER0R ST	
	rw		rw	rw		rw				rw				rw	

Bits	Fields	Descriptions
31	RFRST	RF reset This bit is set and reset by software. 0: No reset 1: Reset the RF
30	HPDFRST	HPDF reset This bit is set and reset by software. 0: No reset 1: Reset the HPDF
29:19	Reserved	Must be kept at reset value.
18	TIMER16 RST	TIMER16 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER16
17	TIMER15 RST	TIMER15 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER15

16:15	Reserved	Must be kept at reset value.
14	SYSCFGRST	<p>SYSCFG reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the SYSCFG</p>
13	Reserved	Must be kept at reset value.
12	SPI0RST	<p>SPI0 reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the SPI0</p>
11	SDIORST	<p>SDIO reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the SDIO</p>
10:9	Reserved	Must be kept at reset value.
8	ADC0RST	<p>ADC0 reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the ADC0</p>
7:5	Reserved	Must be kept at reset value.
4	USART2RST	<p>USART2 reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the USART2</p>
3:1	Reserved	Must be kept at reset value.
0	TIMER0RST	<p>TIMER0 reset</p> <p>This bit is set and reset by software.</p> <p>0: No reset</p> <p>1: Reset the TIMER0</p>

6.5.10. AHB1 enable register (RCU_AHB1EN)

Address offset: 0x30

Reset value: 0x000F 0080

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	USBFSE N	Reserved				DMA1EN	DMA0EN	Reserved	SRAM3E N	SRAM2E N	SRAM1E N	SRAM0E N
		rw				rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3
Reserved	WIFIRUN EN	WIFIEN	CRCEN	Reserved		TSIEN	TZGPCE N	Reserved				PCEN
	rw	rw	rw			rw	rw					rw
2	1	0										
19	18	17	16	15	14	13	12	11	10	9	8	7
19	18	17	16	15	14	13	12	11	10	9	8	7

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	USBFSEN	<p>USBFS clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled USBFS clock</p> <p>1: Enabled USBFS clock</p>
28:23	Reserved	Must be kept at reset value.
22	DMA1EN	<p>DMA1 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled DMA1 clock</p> <p>1: Enabled DMA1 clock</p>
21	DMA0EN	<p>DMA0 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled DMA0 clock</p> <p>1: Enabled DMA0 clock</p>
20	Reserved	Must be kept at reset value.
19	SRAM3EN	<p>SRAM3 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SRAM3 clock</p> <p>1: Enabled SRAM3 clock</p>
18	SRAM2EN	<p>SRAM2 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SRAM2 clock</p> <p>1: Enabled SRAM2 clock</p>
17	SRAM1EN	<p>SRAM1 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SRAM1 clock</p> <p>1: Enabled SRAM1 clock</p>
16	SRAM0EN	<p>SRAM0 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SRAM0 clock</p>

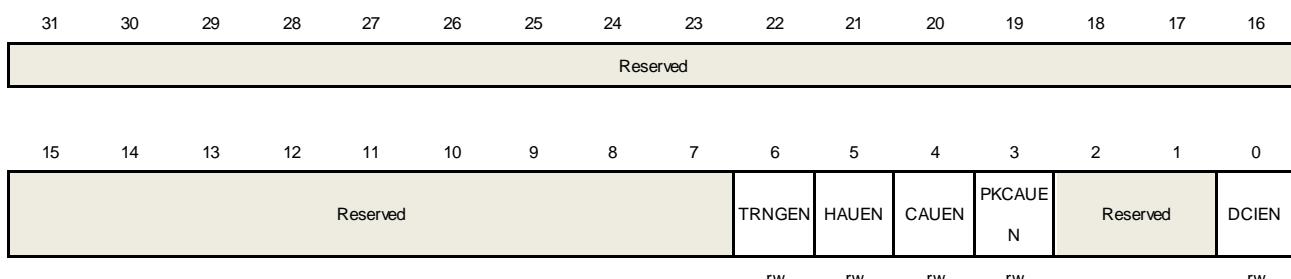
		1: Enabled SRAM0 clock
15	Reserved	Must be kept at reset value
14	WIFIRUNEN	<p>WIFIRUNEN clock enable, This bit is set and reset by software. If WIFIEN is 0, this bit don't work.</p> <p>0: Disabled WIFIRUNEN clock 1: Enabled WIFIRUNEN clock</p>
13	WIFIEN	<p>Wi-Fi Module clock enable This bit is set and reset by software.</p> <p>0: Disabled Wi-Fi clock 1: Enabled Wi-Fi clock</p>
12	CRCEN	<p>CRC clock enable This bit is set and reset by software.</p> <p>0: Disabled CRC clock 1: Enabled CRC clock</p>
11:9	Reserved	Must be kept at reset value.
8	TSIEN	<p>TSI clock enable This bit is set and reset by software.</p> <p>0: Disabled TSI clock 1: Enabled TSI clock</p>
7	TZGPCEN	<p>TZGPC clock enable This bit is set and reset by software.</p> <p>0: Disabled TZGPC clock 1: Enabled TZGPC clock</p>
6:3	Reserved	Must be kept at reset value
2	PCEN	<p>GPIO port C clock enable This bit is set and reset by software.</p> <p>0: Disabled GPIO port C clock 1: Enabled GPIO port C clock</p>
1	PBEN	<p>GPIO port B clock enable This bit is set and reset by software.</p> <p>0: Disabled GPIO port B clock 1: Enabled GPIO port B clock</p>
0	PAEN	<p>GPIO port A clock enable This bit is set and reset by software.</p> <p>0: Disabled GPIO port A clock 1: Enabled GPIO port A clock</p>

6.5.11. AHB2 enable register (RCU_AHB2EN)

Address offset: 0x34

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRNGEN	<p>TRNG clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled TRNG clock</p> <p>1: Enabled TRNG clock</p>
5	HAUEN	<p>HAU clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled HAU clock</p> <p>1: Enabled HAU clock</p>
4	CAUEN	<p>CAU clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled CAU clock</p> <p>1: Enabled CAU clock</p>
3	PKCAUEN	<p>PKCAU clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled PKCAU clock</p> <p>1: Enabled PKCAU clock</p>
2:1	Reserved	Must be kept at reset value.
0	DCIEN	<p>DCI clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled DCI clock</p> <p>1: Enabled DCI clock</p>

6.5.12. AHB3 enable register (RCU_AHB3EN)

Address offset: 0x38

Reset value: 0x0000 0000

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

rw rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QSPIEN	QSPI clock enable This bit is set and reset by software. 0: Disabled QSPI clock 1: Enabled QSPI clock
0	SQPIEN	SQPI clock enable This bit is set and reset by software. 0: Disabled SQPI clock 1: Enabled SQPI clock

6.5.13. APB1 enable register (RCU_APB1EN)

Address offset: 0x40

Reset value: 0x0000 0000

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	PMUEN	Reserved			I2C1EN	I2C0EN	Reserved		USART0EN	USART1EN	Reserved				
		rw					rw	rw			rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1EN	Reserved	WWDTGTEN	Reserved					TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN		
									rw	rw	rw	rw	rw		

		This bit is set and reset by software.
		0: Disabled PMU clock
		1: Enabled PMU clock
27:23	Reserved	Must be kept at reset value.
22	I2C1EN	I2C1 clock enable
		This bit is set and reset by software.
		0: Disabled I2C1 clock
		1: Enabled I2C1 clock
21	I2C0EN	I2C0 clock enable
		This bit is set and reset by software.
		0: Disabled I2C0 clock
		1: Enabled I2C0 clock
20:19	Reserved	Must be kept at reset value
18	USART0EN	USART0 clock enable
		This bit is set and reset by software.
		0: Disabled USART0 clock
		1: Enabled USART0 clock
17	USART1EN	USART1 clock enable
		This bit is set and reset by software.
		0: Disabled USART1 clock
		1: Enabled USART1 clock
16:15	Reserved	Must be kept at reset value.
14	SPI1EN	SPI1 clock enable
		This bit is set and reset by software.
		0: Disabled SPI1 clock
		1: Enabled SPI1 clock
13:12	Reserved	Must be kept at reset value.
11	WWDGTEN	WWDGT clock enable
		This bit is set and reset by software.
		0: Disabled WWDGT clock
		1: Enabled WWDGT clock
10:5	Reserved	Must be kept at reset value.
4	TIMER5EN	TIMER5 clock enable
		This bit is set and reset by software.
		0: Disabled TIMER5 clock
		1: Enabled TIMER5 clock
3	TIMER4EN	TIMER4 clock enable

		This bit is set and reset by softw are.
		0: Disabled TIMER4 clock
		1: Enabled TIMER4 clock
2	TIMER3EN	TIMER3 clock enable
		This bit is set and reset by softw are.
		0: Disabled TIMER3 clock
		1: Enabled TIMER3 clock
1	TIMER2EN	TIMER2 clock enable
		This bit is set and reset by softw are.
		0: Disabled TIMER2 clock
		1: Enabled TIMER2 clock
0	TIMER1EN	TIMER1 clock enable
		This bit is set and reset by softw are.
		0: Disabled TIMER1 clock
		1: Enabled TIMER1 clock

6.5.14. APB2 enable register (RCU_APB2EN)

Address offset: 0x44

Reset value: 0x0000 0000

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
RFEN	HPDFEN												TIMER16 EN	TIMER15 EN	Reserved
rw	rw												rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYSCFG EN	Reserved	SPI0EN	SDIOEN	Reserved	ADC0EN		Reserved		USART2 EN		Reserved		TIMER0E N	
	rw		rw	rw		rw				rw				rw	

Bits	Fields	Descriptions
31	RFEN	RF clock enable This bit is set and reset by softw are. 0: Disabled RF clock 1: Enabled RF clock
30	HPDFEN	HPDF clock enable This bit is set and reset by softw are. 0: Disabled HPDF clock 1: Enabled HPDF clock

29:19	Reserved	Must be kept at reset value.
18	TIMER16EN	<p>TIMER16 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled TIMER16 clock</p> <p>1: Enabled TIMER16 clock</p>
17	TIMER15EN	<p>TIMER15 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled TIMER15 clock</p> <p>1: Enabled TIMER15 clock</p>
16:15	Reserved	Must be kept at reset value.
14	SYSCFGEN	<p>SYSCFG clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SYSCFG clock</p> <p>1: Enabled SYSCFG clock</p>
13	Reserved	Must be kept at reset value.
12	SPI0EN	<p>SPI0 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SPI0 clock</p> <p>1: Enabled SPI0 clock</p>
11	SDIOEN	<p>SDIO clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SDIO clock</p> <p>1: Enabled SDIO clock</p>
10:9	Reserved	Must be kept at reset value.
8	ADC0EN	<p>ADC0 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled ADC0 clock</p> <p>1: Enabled ADC0 clock</p>
7:5	Reserved	Must be kept at reset value.
4	USART2EN	<p>USART2 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled USART2 clock</p> <p>1: Enabled USART2 clock</p>
3:1	Reserved	Must be kept at reset value.
0	TIMEROEN	<p>TIMER0 clock enable</p> <p>This bit is set and reset by software.</p> <p>0: Disabled TIMER0 clock</p>

1: Enabled TIMERO clock

6.5.15. AHB1 sleep mode enable register (RCU_AHB1SPEN)

Address offset: 0x50

Reset value: 0x206F F187

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	USBFSS PEN	Reserved				DMA1SP EN	DMA0SP EN	Reserved	SRAM3S PEN	SRAM2S PEN	SRAM1S PEN	SRAM0S PEN			
		rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCSPEN	WIFIRUN	WIFISPE	CRCSPEN	Reserved		TSISPEN	TZGPCS PEN	Reserved				PCSPEN	PBSPEN	PASPEN	
rw	rw	rw	rw			rw	rw					rw	rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	USBFSSPEN	USBFS clock enable when sleep mode This bit is set and reset by software. 0: Disabled USBFS clock when sleep mode 1: Enabled USBFS clock when sleep mode
28:23	Reserved	Must be kept at reset value.
22	DMA1SPEN	DMA1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled DMA1 clock when sleep mode 1: Enabled DMA1 clock when sleep mode
21	DMA0SPEN	DMA0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled DMA0 clock when sleep mode 1: Enabled DMA0 clock when sleep mode
20	Reserved	Must be kept at reset value.
19	SRAM3SPEN	SRAM3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SRAM3 clock when sleep mode 1: Enabled SRAM3 clock when sleep mode
18	SRAM2SPEN	SRAM2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SRAM2 clock when sleep mode

		1: Enabled SRAM2 clock when sleep mode
17	SRAM1SPEN	<p>SRAM1 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled SRAM1 clock when sleep mode
		1: Enabled SRAM1 clock when sleep mode
16	SRAM0SPEN	<p>SRAM0 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled SRAM0 clock when sleep mode
		1: Enabled SRAM0 clock when sleep mode
15	FMCSPEN	<p>FMC clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled FMC clock when sleep mode
		1: Enabled FMC clock when sleep mode
14	WIFIRUNSPEN	<p>WIFIRUN clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled WIFIRUN clock when sleep mode
		1: Enabled WIFIRUN clock when sleep mode
13	WIFISPEN	<p>Wi-Fi clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled Wi-Fi clock when sleep mode
		1: Enabled Wi-Fi clock when sleep mode
12	CRCSPEN	<p>CRC clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled CRC clock when sleep mode
		1: Enabled CRC clock when sleep mode
11:9	Reserved	Must be kept at reset value.
8	TSISPEN	<p>TSI clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled TSI clock when sleep mode
		1: Enabled TSI clock when sleep mode
7	TZGPCSPEN	<p>TZGPC clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled TZGPC clock when sleep mode
		1: Enabled TZGPC clock when sleep mode
6:3	Reserved	Must be kept at reset value.
2	PCSPEN	<p>GPIO port C clock enable when sleep mode</p> <p>This bit is set and reset by software.</p>
		0: Disabled GPIO port C clock when sleep mode

		1: Enabled GPIO port C clock when sleep mode
1	PBSPEN	GPIO port B clock enable when sleep mode This bit is set and reset by software.
		0: Disabled GPIO port B clock when sleep mode
		1: Enabled GPIO port B clock when sleep mode
0	PASPEN	GPIO port A clock enable when sleep mode This bit is set and reset by software.
		0: Disabled GPIO port A clock when sleep mode
		1: Enabled GPIO port A clock when sleep mode

6.5.16. AHB2 sleep mode enable register (RCU_AHB2SPEN)

Address offset: 0x54

Reset value: 0x0000 0079

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TRNGSP EN	HAUSPE N	CAUSPE N	PKCAUS PEN	Reserved		DCISPEN	
								rw	rw	rw	rw			rw	

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRNGSPEN	TRNG clock enable when sleep mode This bit is set and reset by software. 0: Disabled TRNG clock when sleep mode 1: Enabled TRNG clock when sleep mode
5	HAUSPEN	HAU clock enable when sleep mode This bit is set and reset by software. 0: Disabled HAU clock when sleep mode 1: Enabled HAU clock when sleep mode
4	CAUSPEN	CAU clock enable when sleep mode This bit is set and reset by software. 0: Disabled CAU clock when sleep mode 1: Enabled CAU clock when sleep mode
3	PKCAUSPEN	PKCAU clock enable when sleep mode This bit is set and reset by software.

		0: Disabled PKCAU clock when sleep mode 1: Enabled PKCAU clock when sleep mode
2:1	Reserved	Must be kept at reset value.
0	DCISPEN	DCI clock enable when sleep mode This bit is set and reset by software. 0: Disabled DCI clock when sleep mode 1: Enabled DCI clock when sleep mode

6.5.17. AHB3 sleep mode enable register (RCU_AHB3SPEN)

Address offset: 0x58

Reset value: 0x0000 0003

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

rw rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QSPISPEN	QSPI clock enable when sleep mode This bit is set and reset by software. 0: Disabled QSPI clock when sleep mode 1: Enabled QSPI clock when sleep mode
0	SQPISPEN	SQPI clock enable when sleep mode This bit is set and reset by software. 0: Disabled SQPI clock when sleep mode 1: Enabled SQPI clock when sleep mode

6.5.18. APB1 sleep mode enable register (RCU_APB1SPEN)

Address offset: 0x60

Reset value: 0x1066 481F

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		PMUSPE N	Reserved				I2C1SPE N	I2C0SPE N	Reserved		USART0 SPEN	USART1 SPEN	Reserved		
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1SPE N	Reserved	WWDTGT SPEN	Reserved				TIMER5S PEN	TIMER4S PEN	TIMER3S PEN	TIMER2S PEN	TIMER1S PEN			
	rw		rw					rw		rw		rw		rw	

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	PMUSPEN	<p>PMU clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled PMU clock when sleep mode</p> <p>1: Enabled PMU clock when sleep mode</p>
27:23	Reserved	Must be kept at reset value.
22	I2C1SPEN	<p>I2C1 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled I2C1 clock when sleep mode</p> <p>1: Enabled I2C1 clock when sleep mode</p>
21	I2C0SPEN	<p>I2C0 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled I2C0 clock when sleep mode</p> <p>1: Enabled I2C0 clock when sleep mode</p>
20:19	Reserved	Must be kept at reset value
18	USART0SPEN	<p>USART0 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled USART0 clock when sleep mode</p> <p>1: Enabled USART0 clock when sleep mode</p>
17	USART1SPEN	<p>USART1 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled USART1 clock when sleep mode</p> <p>1: Enabled USART1 clock when sleep mode</p>
16:15	Reserved	Must be kept at reset value.
14	SPI1SPEN	<p>SPI1 clock enable when sleep mode</p> <p>This bit is set and reset by software.</p> <p>0: Disabled SPI1 clock when sleep mode</p> <p>1: Enabled SPI1 clock when sleep mode</p>
13:12	Reserved	Must be kept at reset value.

11	WWDGTS PEN	WWDG T clock enable when sleep mode This bit is set and reset by software. 0: Disabled WWDGT clock when sleep mode 1: Enabled WWDGT clock when sleep mode
10:5	Reserved	Must be kept at reset value.
4	TIMER5SPEN	TIMER5 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER5 clock when sleep mode 1: Enabled TIMER5 clock when sleep mode
3	TIMER4SPEN	TIMER4 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER4 clock when sleep mode 1: Enabled TIMER4 clock when sleep mode
2	TIMER3SPEN	TIMER3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER3 clock when sleep mode 1: Enabled TIMER3 clock when sleep mode
1	TIMER2SPEN	TIMER2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER2 clock when sleep mode 1: Enabled TIMER2 clock when sleep mode
0	TIMER1SPEN	TIMER1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER1 clock when sleep mode 1: Enabled TIMER1 clock when sleep mode

6.5.19. APB2 sleep mode enable register (RCU_APB2SPEN)

Address offset: 0x64

Reset value: 0xC006 5911

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
RFSPEN	HPDFDS PEN	Reserved										TIMER16 SPEN	TIMER15 SPEN	Reserved	
rw	rw											rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYSCFG SPEN	Reserved	SPIOSPE	SDIOSPE	Reserved	ADCOSPE EN	Reserved	Reserved	USART2 SPEN	Reserved	Reserved	Reserved	Reserved	TIMER0S PEN	
	rw		rw	rw		rw			rw			rw		rw	

Bits	Fields	Descriptions
31	RFSPEN	RF clock enable when sleep mode This bit is set and reset by software. 0: Disabled RF clock when sleep mode 1: Enabled RF clock when sleep mode
30	HPDFSPEN	HPDF clock enable when sleep mode This bit is set and reset by software. 0: Disabled HPDF clock when sleep mode 1: Enabled HPDF clock when sleep mode
29:19	Reserved	Must be kept at reset value.
18	TIMER16SPEN	TIMER16 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER16 clock when sleep mode 1: Enabled TIMER16 clock when sleep mode
17	TIMER15SPEN	TIMER15 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER15 clock when sleep mode 1: Enabled TIMER15 clock when sleep mode
16:15	Reserved	Must be kept at reset value.
14	SYSCFGSPEN	SYSCFG clock enable when sleep mode This bit is set and reset by software. 0: Disabled SYSCFG clock when sleep mode 1: Enabled SYSCFG clock when sleep mode
13	Reserved	Must be kept at reset value
12	SPI0SPEN	SPI0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI0 clock when sleep mode 1: Enabled SPI0 clock when sleep mode
11	SDIOSPEN	SDIO clock enable when sleep mode This bit is set and reset by software. 0: Disabled SDIO clock when sleep mode 1: Enabled SDIO clock when sleep mode
10:9	Reserved	Must be kept at reset value.
8	ADC0SPEN	ADC0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled ADC0 clock when sleep mode 1: Enabled ADC0 clock when sleep mode

7:5	Reserved	Must be kept at reset value.
4	USART2SPEN	USART2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USART2 clock when sleep mode 1: Enabled USART2 clock when sleep mode
3:1	Reserved	Must be kept at reset value.
0	TIMER0SPEN	TIMER0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER0 clock when sleep mode 1: Enabled TIMER0 clock when sleep mode

6.5.20. Backup domain control register (RCU_BDCTL)

Address offset: 0x70

Reset value: 0x0000 0018, reset by Backup domain Reset.

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

Note: The LXTALEN, LXTALBPS, RTCSRC and RTCEN bits of the Backup domain control register (RCU_BDCTL) are only reset after a Backup domain Reset. These bits can be modified only when the BKPRST bit in the Power control register (PMU_CTL) is set.

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup domain reset This bit is set and reset by software. 0: No reset 1: Resets Backup domain
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock 1: Enabled RTC clock
14:10	Reserved	Must be kept at reset value.

9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. Once the RTC clock source has been selected, it cannot be changed anymore unless the Backup domain is reset. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC32K selected as RTC source clock 11: (CK_HXTAL / RTCDIV) selected as RTC source clock, please refer to RTCDIV bits in RCU_CFG0 register.
7:5	Reserved	Must be kept at reset value.
4:3	LXTALDRI[1:0]	LXTAL drive capability Set and reset by software. Backup domain reset resets this value. 00: low er driving capability 01: high driving capability 10: higher driving capability 11: highest driving capability (reset value) Note: The LXTALDRI is not in bypass mode.
2	LXTALBPS	LXTAL bypass mode enable Set and reset by software. 0: Disable the LXTAL Bypass mode 1: Enable the LXTAL Bypass mode
1	LXTALSTB	Low speed crystal oscillator stabilization flag Set by hardware to indicate if the LXTAL output clock is stable and ready for use. 0: LXTAL is not stable 1: LXTAL is stable
0	LXTALEN	LXTAL enable Set and reset by software. 0: Disable LXTAL 1: Enable LXTAL

6.5.21. Reset source/clock register (RCU_RSTSCK)

Address offset: 0x74

Reset value: 0x0C00 0000, all reset flags reset by power reset only, RSTFC/IRC32KEN reset by system reset.

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	WWDGTRSTF	FWDGTRSTF	SW RSTF	POR RSTF	EP RSTF	OBLRST F	RSTFC	Reserved							

r r r r r r r rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IRC32K STB	IRC32KE N
														r	rw

Bits	Fields	Descriptions
31	LPRSTF	<p>Low -power reset flag</p> <p>Set by hardware when Deep-sleep /standby reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No Low -power management reset generated</p> <p>1: Low -power management reset generated</p>
30	WWDGTRSTF	<p>Window watchdog timer reset flag</p> <p>Set by hardware when a window watchdog timer reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No window watchdog reset generated</p> <p>1: Window watchdog reset generated</p>
29	FWDGTRSTF	<p>Free watchdog timer reset flag</p> <p>Set by hardware when a free watchdog timer reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No free watchdog timer reset generated</p> <p>1: free Watchdog timer reset generated</p>
28	SWRSTF	<p>Software reset flag</p> <p>Set by hardware when a software reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No software reset generated</p> <p>1: Software reset generated</p>
27	PORRSTF	<p>Power reset flag</p> <p>Set by hardware when a Power reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No Power reset generated</p> <p>1: Power reset generated</p>
26	EPRSTF	<p>External PIN reset flag</p> <p>Set by hardware when an External PIN reset generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No External PIN reset generated</p> <p>1: External PIN reset generated</p>
25	OBLRSTF	<p>Option byte loader reset flag</p> <p>Set by hardware when an option byte loader generated.</p> <p>Reset by writing 1 to the RSTFC bit.</p> <p>0: No Option byte loader reset generated</p>

		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
23:2	Reserved	Must be kept at reset value.
1	IRC32KSTB	IRC32K stabilization flag Set by hardware to indicate if the IRC32K output clock is stable and ready for use. 0: IRC32K is not stable 1: IRC32K is stable
0	IRC32KEN	IRC32K enable Set and reset by software. 0: Disable IRC32K 1: Enable IRC32K

6.5.22. PLL clock spread spectrum control register (RCU_PLLSSCTL)

Address offset: 0x80

Reset value: 0x0000 0000

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

The spread spectrum modulation is available only for the main PLL clock

The RCU_PLLSSCTL register must be written when the main PLL is disabled

This register is used to configure the PLL spread spectrum clock generation according to the following formulas:

$$\text{MODCNT} = \text{round}(f_{\text{PLLIN}}/4/f_{\text{mod}})$$

$$\text{MODSTEP} = \text{round}(\text{mdamp} * \text{PLLN} * 2^{14} / (\text{MODCNT} * 100))$$

Where f_{PLLIN} represents the PLL input clock frequency, f_{mod} represents the spread spectrum modulation frequency, mdamp represents the spread spectrum modulation amplitude expressed as a percentage, PLLN represents the PLL clock frequency multiplication factor.

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

Bits	Fields	Descriptions
31	SSCGON	PLL spread spectrum modulation enable 0: Spread spectrum modulation disable 1: Spread spectrum modulation enable

30	SS_TYPE	PLL spread spectrum modulation type select 0: Center spread selected 1: Down spread selected
29:28	Reserved	Must be kept at reset value.
27:13	MODSTEP[14:0]	These bits configure PLL spread spectrum modulation profile amplitude and frequency. The following criteria must be met: MODSTEP*MODCNT≤2 ¹⁶ -1
12:0	MODCNT[12:0]	These bits configure PLL spread spectrum modulation profile amplitude and frequency. The following criteria must be met: MODSTEP*MODCNT≤2 ¹⁶ -1

6.5.23. PLL clock configuration register (RCU_PLLCFG)

Address offset: 0x84

Reset value: 0x0300 0000

To configure the PLLI2S clock, refer to the following formula:

$$CK_{PLLI2SVCOSRC} = CK_{PLLI2SSRC} / PLLI2SPSC$$

$$CK_{PLLI2SVCO} = CK_{PLLI2SVCOSRC} \times PLLI2SN$$

$$CK_{PLLI2S} = CK_{PLLI2SVCO} / PLLI2SDIV$$

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
PLLDIGFSYSDIV[5:0]				PLLDIGOSEL[1:0]		Reserved				PLLI2SPSC[2:0]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLLI2SN[6:0]				Reserved		PLLI2SDIV[5:0]				rw				

Bits	Fields	Descriptions
31:26	PLLDIGFSYSDIV[5:0]	PLLDIG clock divider factor for system clock Set and reset by software to control the PLLDIG clock divider factor for system clock. 000000: PLLDIG clock divided by 1 for system clock 000001: PLLDIG clock divided by 2 for system clock 000010: PLLDIG clock divided by 2 for system clock ... 111111: PLLDIG clock divided by 64 for system clock
25:24	PLLDIGOSEL[1:0]	PLLDIG output frequency select 00: selected 192Mhz as PLLDIG output frequency 01: selected 240Mhz as PLLDIG output frequency 10: selected 320Mhz as PLLDIG output frequency. 11: selected 480Mhz as PLLDIG output frequency

23:19	Reserved	Must be kept at reset value.
18:16	PLLl2SPSC[2:0]	<p>The PLLl2S VCO source clock pre-scale.</p> <p>Set and reset by software when the PLLl2S is disable. These bits used to generate the clock of PLLl2SVCO source clock CK_PLLl2SV COSRC.</p> <p>000: CK_PLLl2SSRC / 1</p> <p>001: CK_PLLl2SSRC / 2</p> <p>.....</p> <p>110: CK_PLLl2SSRC / 7</p> <p>111: CK_PLLl2SSRC / 8</p>
15	Reserved	Must be kept at reset value.
14:8	PLLl2SN[8:0]	<p>The PLLl2S VCO clock multiplication factor</p> <p>Set and reset by software, when the PLLl2S is disable. These bits used to generate PLLl2S VCO clock (CK_PLLl2SV CO) from PLLl2S VCO source clock (CK_PLLl2SV COSRC).</p> <p>Note: The frequency of CK_PLLl2SV CO is between 64MHz to 500MHz</p> <p>The value of PLLl2SN must : $8 \leq \text{PLLl2SN} \leq 127$</p> <p>0000000: Reserved</p> <p>0000001: Reserved</p> <p>...</p> <p>0000111: Reserved</p> <p>0001000: CK_PLLl2SVCO = CK_PLLl2SV COSRC x 8.</p> <p>0001001: CK_PLLl2SVCO = CK_PLLl2SV COSRC x 9.</p> <p>...</p> <p>1111111: CK_PLLl2SVCO = CK_PLLl2SV COSRC x 127</p>
7:6	Reserved	Must be kept at reset value.
5:0	PLLl2SDIV[5:0]	<p>PLLl2S clock divider factor.</p> <p>This bit is set and reset by software.</p> <p>000000: PLLl2SDIV input source clock divided by 32</p> <p>000001: Reserved</p> <p>000010: Reserved</p> <p>000011: PLLl2SDIV input source clock divided by 1.5</p> <p>000100: PLLl2SDIV input source clock divided by 2</p> <p>000101: PLLl2SDIV input source clock divided by 2.5</p> <p>...</p> <p>111111: PLLl2SDIV input source clock divided by 31.5</p>

6.5.24. Clock configuration register 1 (RCU_CFG1)

Address offset: 0x8C

Reset value: 0x0030 0600

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
USART0SEL[1:0]	USART2SEL[1:0]	I2C0SEL[1:0]	Reserved	TIMERSEL	Reserved	LDO_ANA_LQB	LDO_CLK_LQB	BGPU	LDOCLK_PU	LDOANA_PU	RFPLLPU				
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
RFPLLLO CK	RFPLLCA LEN	Reserved	BGVBIT[2:0]									IRC16MDIV[8:0]			
RO	RW		RW									RW			

Bits	Fields	Descriptions
31:30	USART0SEL[1:0]	USART0 Clock Source Selection Set and reset by softw are to control the USART0 clock source. 00: CK_APB1 selected as USART0 source clock 01: CK_SYS selected as USART0 source clock 10: CK_LXTAL selected as USART0 source clock 11: CK_IRC16M selected as USART0 source clock
29:28	USART2SEL[1:0]	USART2 Clock Source Selection Set and reset by softw are to control the USART1 clock source. 00: CK_APB2 selected as USART2 source clock 01: CK_SYS selected as USART2 source clock 10: CK_LXTAL selected as USART2 source clock 11: CK_IRC16M selected as USART2 source clock
27:26	I2C0SEL[1:0]	I2C0 Clock Source Selection Set and reset by softw are to control the I2C0 clock source. 00: CK_APB1 selected as I2C0 source clock 01: CK_SYS selected as I2C0 source clock 1x: CK_IRC16M selected as I2C0 source clock
25	Reserved	Must be kept at reset value.
24	TIMERSEL	TIMER clock selection This bit is set and reset by softw are. This bit defined all timer clock selection. 0: If APB1PSC/APB2PSC in RCU_CFG0 register is 0b0xx(CK_APBx = CK_AHB) or 0b100(CK_APBx = CK_AHB/2), the TIMER clock is equal to CK_AHB(CK_TIMERx = CK_AHB). Or else, the TIMER clock is twice the corresponding APB clock (TIMER in APB1 domain: CK_TIMERx = 2 x CK_APB1; TIMER in APB2 domain: CK_TIMERx = 2 x CK_APB2). 1: If APB1PSC/APB2PSC in RCU_CFG0 register is 0b0xx(CK_APBx = CK_AHB), 0b100(CK_APBx = CK_AHB/2), or 0b101(CK_APBx = CK_AHB/4), the TIMER clock is equal to CK_AHB(CK_TIMERx = CK_AHB). Or else, the TIMER clock is four times the corresponding APB clock (TIMER in APB1 domain: CK_TIMERx = 4 x CK_APB1, TIMER in APB2 domain: CK_TIMERx = 4 x CK_APB2).

23:22	Reserved	Must be kept at reset value
21	LDO_ANA_LQB	Trust zone security by RF 0: Analog LDO high current bias mode 1: Analog LDO low current bias mode
20	LDO_CLK_LQB	Trust zone security by RF 0: Clock LDO high current bias mode 1:Clock LDO low current bias mode
19	BGPU	BandGap power on enable, Trust zone security by RF When PLLDIGEN is 1, this bit can't be written 0 0: BandGap power down 1: BandGap power on
18	LDOCLKPU	LDO clock power on enable for RF/ADC/DAC, Trust zone security by RF 0: LDO clock power down 1: LDO clock power on
17	LDOANAPU	LDO analog power on enable for RF filter, Trust zone security by RF 0: LDO analog power down 1: LDO analog power on
16	RFPLLPU	RFPLL power on enable, Trust zone security by RF 0: RFPLL power down 1: RFPLL power on
15	RFPLLLOCK	RF PLL LOCK. 0: RFPLL is not Lock 1: RFPLL locked
14	RFPLLCALEN	RF PLL Calculation enable. Trust zone security by RF 0: RF PLL Calculation disable 1: RF PLL Calculation enable
13:12	Reserved	Must be kept at reset value
11:9	BGVBIT[2:0]	BandGap Power adjust, which can't be written when HXTALON, HXTALEN2S, HXTALENPLL or PLLDIGEN is on. Trust zone security by RF
8:0	IRC16MDIV[8:0]	IRC16M clock divider factor for system clock. It can't be written when system clock select IRC16M or IRC16MEN. 00000000: IRC16M clock divided by 1 00000001: IRC16M clock divided by 2 00000010: IRC16M clock divided by 3 ... 11111111: IRC16M clock divided by 512

6.5.25. Additional clock control register (RCU_ADDCTL)

Address offset: 0x90

Reset value: 0x0000 0000

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					PLLFI2SDIV[5:0]				Reserved	SDIOSEL[1]	SDIODIV[5:0]				SDIOSEL[0]
					rw				rw		rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPDFAUDIOSEL[1:0]		I2SSEL[1:0]	HPDFSEL		Reserved				USBFSDIV[4:0]				USBFSS		EL
	rw	rw	rw							rw			rw		rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:24	PLLFI2SDIV[5:0]	<p>The PLL divider factor for I2S clock</p> <p>Set and reset by softw are to control the PLL clock divider factor for I2S clock.</p> <p>000000: PLL clock divided by 1 for I2S clock</p> <p>000001: PLL clock divided by 2 for I2S clock</p> <p>...</p> <p>111111: PLL clock divided by 64 for I2S clock</p>
23	Reserved	Must be kept at reset value.
22	SDIOSEL[1]	<p>Bit 1 of SDIOSEL</p> <p>See bits 16 of. RCU_ADDCTL</p>
21:17	SDIODIV[4:0]	<p>SDIO clock divider factor</p> <p>This bit is set and reset by softw are.</p> <p>00000: SDIODIV input source clock divided by 1</p> <p>00001: SDIODIV input source clock divided by 2</p> <p>...</p> <p>11111: SDIODIV input source clock divided by 32</p>
16	SDIOSEL[0]	<p>SDIO clock selection</p> <p>Set and reset by softw are. This bit used to generate SDIO clock w hich select PLL or PLLDIG clock.</p> <p>00: PLL clock select as SDIO source clock</p> <p>01: PLLDIG clock select as SDIO source clock</p> <p>10: IRC16M selected as SDIO source clock</p> <p>11: HXTAL selected as SDIO source clock</p>
15:14	HPDFAUDIOSEL[1:0]	HPDF AUDIO clock Source Selection

	[]	Set and reset by softw are to control the HPDF_AUDIO clock source. 00: PLLI2S output clock selected as HPDF_AUDIO source clock 01: External I2S_CKIN PIN selected as HPDF_AUDIO source clock 10: PLL division selected as HPDF_AUDIO source clock 11: IRC16M selected as HPDF_AUDIO source clock
13:12	I2SSEL[1:0]	I2S Clock Source Selection Set and reset by softw are to control the I2S clock source. 00: PLLI2S output clock selected as I2S source clock 01: External I2S_CKIN PIN selected as I2S source clock 10: PLL division selected as I2S source clock 11: Reserved
11	HPDFSEL	HPDF clock Source Selection Set and reset by softw are to control the HPDF clock source. 0: PCLK2 Clock selected as HPDF source clock 1: System Clock selected as HPDF source clock
10:6	Reserved	Must be kept at reset value.
5:1	USBFSDIV[4:0]	USBFS clock divider factor Division the PLL or PLLDIG clock for USBFS clock, according to USBFSSEL bit. 00000: USBFSDIV input source clock divided by 1 00001: USBFSDIV input source clock divided by 2 ... 11111: USBFSDIV input source clock divided by 32
0	USBFSSEL	USBFS clock selection Set and reset by softw are. This bit used to generate USBFS clock which select PLL or PLLDIG clock. 0: PLL clock select as USBFS source clock 1: PLLDIG clock select as USBFS source clock

6.5.26. Secure protection configuration register (RCU_SECP_CFG)

Address offset: 0xC0

Reset value: 0x0000 0000

When TZEN = 1, this register provides a write access security configuration, which can be referenced [Table 6-4. RCU secure protection configuration summary](#). When TZEN = 0, this register is RAZ/WI.

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Reserved	BKPSEC P	RMVRST FSECP	PLL2SS ECP	PLLDIGS ECP	PLLSECP ECP	PRESCS ECP	SYSCLK SECP	LXTALSE CP	IRC32KS ECP	HXTALS ECP	IRC16MS ECP
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	BKPSEC P	BKP security protection Set and reset by softw are. 0: Non secure 1: Secure
9	RMVRSTFSECP	Remove reset flag security protection Set and reset by softw are. 0: Non secure 1: Secure
8	PLL2SSSEC P	PLL2S configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure
7	PLLDIGSEC P	PLLDIG configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure
6	PLLSECP	Main PLL configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure
5	PRESCS ECP	AHBx/APBx prescaler configuration bits security protection Set and reset by softw are. 0: Non secure 1: Secure
4	SYSCLKSEC P	SYSCLK clock selection, clock output on MCO configuration Set and reset by softw are. 0: Non secure 1: Secure
3	LXTALSEC P	LXTAL clock configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure

2	IRC32KSECP	IRC32K clock configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure
1	HXTALSECP	HXTAL clock configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure
0	IRC16MSECP	IRC16M clock configuration and status bits security protection Set and reset by softw are. 0: Non secure 1: Secure

6.5.27. Secure protection status register (RCU_SECP_STAT)

Address offset: 0XC4

Reset value: 0x0000 0000

When TZEN = 1, access is allowed for both privileges and non-privileges. This register provides security status of security configuration bits in RCU_SECP_CFG register. When TZEN = 0, this register is RAZ/WI.

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BKPSEC	RMVFSE	PLL1SS	PLLDIGS	PLLSECP	PRESCS	SYSCLK	LXTALSE	IRC32KS	HXTALS	IRC16MS	
				PF	CF	ECPF	ECPF	F	ECPF	SECPF	CPF	ECPF	ECPF	ECPF	
				r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	BKPSECPF	BKP security protection flag. Set and reset by softw are. 0: Non secure 1: Secure
9	RMVRSTFSECPF	Remove reset flag security protection flag Set and reset by softw are. 0: Non secure 1: Secure

8	PLL12SSEC _{PF}	PLL12S configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
7	PLLDIGSEC _{PF}	PLLDIG configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
6	PLLSEC _{PF}	Main PLL configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
5	PRESCSEC _{PF}	AHBx/APBx prescaler configuration bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
4	SYSCLKSEC _{PF}	SYSCLK clock selection, clock output on MCO configuration Set and reset by softw are. 0: Non secure 1: Secure
3	LXTALSEC _{PF}	LXTAL clock configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
2	IRC32KSEC _{PF}	IRC32K clock configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
1	HXTALSEC _{PF}	HXTAL clock configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure
0	IRC16MSEC _{PF}	IRC16M clock configuration and status bits security protection flag Set and reset by softw are. 0: Non secure 1: Secure

6.5.28. AHB1 secure protection status register (RCU_AHB1SECP_STAT)

Address offset: 0xC8

Reset value: 0x0000 8007 (TZEN = 0)

Reset value: 0x000f 8007 (TZEN = 1)

When TZEN = 1, this register provides AHB1 peripheral clock security status. Privileged and unprivileged, secure and non-secure accesses are all allowed access. When the peripheral is configured to be secure, the corresponding peripheral clock is also secure. When TZEN = 0, this register is RAZ.

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	USBFSS ECPF								DMA1SE CPF	DMA0SE CPF	Reserved	SRAM3S ECPF	SRAM2S ECPF	SRAM1S ECPF	SRAM0S ECPF
	r								r	r		r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCSEC PF	Reserved	WIFISEC PF	CRCSEC PF									PCSECP F	PBSECP F	PASECP F	
r	r	r							r	r		r	r	r	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	USBFSSECF	USBFS security protection flag This flag is set by hardware when it is secure. 0: Non secure USBFS 1: Secure USBFS
28:23	Reserved	Must be kept at reset value
22	DMA1SECF	DMA1 security protection flag This flag is set by hardware when it is secure. 0: Non secure DMA1 1: Secure DMA1
21	DMA0SECF	DMA0 security protection flag This flag is set by hardware when it is secure. 0: Non secure DMA0 1: Secure DMA0
20	Reserved	Must be kept at reset value
19	SRAM3SECF	SRAM3 security protection flag This flag is set by hardware when it is secure. 0: Non secure SRAM3 1: Secure SRAM3
18	SRAM2SECF	SRAM2 security protection flag This flag is set by hardware when it is secure.

		0: Non secure SRAM2 1: Secure SRAM2
17	SRAM1SECF	SRAM1 security protection flag This flag is set by hardware when it is secure. 0: Non secure SRAM1 1: Secure SRAM1
16	SRAM0SECF	SRAM0 security protection flag This flag is set by hardware when it is secure. 0: Non secure SRAM0 1: Secure SRAM0
15	FMCSECF	FMC security protection flag This flag is set by hardware when it is secure.. 0: Non secure FMC 1: Secure FMC
14	Reserved	Must be kept at reset value
13	WIFISECF	Wi-Fi security protection flag This flag is set by hardware when it is secure. 0: Non secure Wi-Fi 1: Secure Wi-Fi
12	CRCSECF	CRC security protection flag This flag is set by hardware when it is secure. 0: Non secure CRC 1: Secure CRC
11:3	Reserved	Must be kept at reset value
2	PCSECF	GPIO port C security protection flag This flag is set by hardware when it is secure. 0: Non secure GPIO port C 1: Secure GPIO port C
1	PBSECF	GPIO port B security protection flag This flag is set by hardware when it is secure. 0: Non secure GPIO port B 1: Secure GPIO port B
0	PASECF	GPIO port A security protection flag This flag is set by hardware when it is secure. 0: Non secure GPIO port A 1: Secure GPIO port A

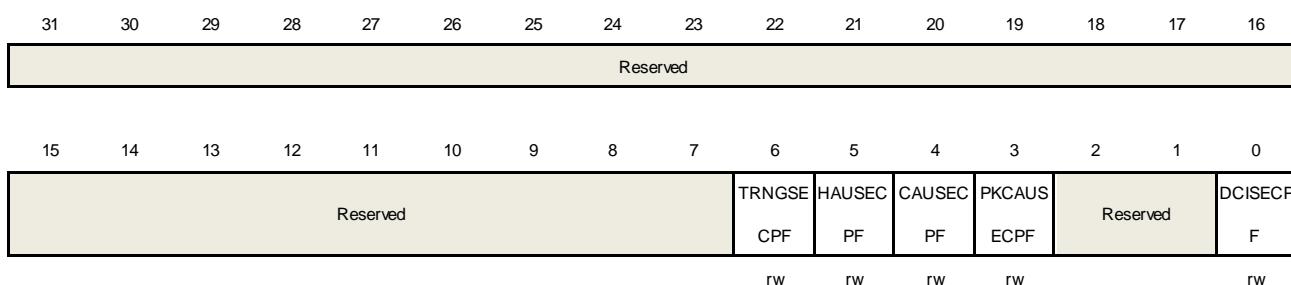
6.5.29. AHB2 secure protection status register (RCU_AHB2SECP_STAT)

Address offset: 0xCC

Reset value: 0x0000 0000

When TZEN = 1, this register provides AHB2 peripheral clock security status. Privileged and unprivileged, secure and non-secure accesses are all allowed access. When the peripheral is configured to be secure, the corresponding peripheral clock is also secure. When TZEN = 0, this register is RAZ.

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	TRNGSECPF	TRNG security protection flag This flag is set by hardware when it is secure. 0: Non secure TRNG 1: Secure TRNG
5	HAUSECPF	HAU security protection flag This flag is set by hardware when it is secure. 0: Non secure HAU 1: Secure HAU
4	CAUSECPF	CAU security protection flag This flag is set by hardware when it is secure. 0: Non secure CAU 1: Secure CAU
3	PKCAUSECPF	PKCAU security protection flag This flag is set by hardware when it is secure. 0: Non secure PKCAU 1: Secure PKCAU
2:1	Reserved	Must be kept at reset value
0	DCISECPF	DCI security protection flag This flag is set by hardware when it is secure. 0: Non secure DCI

1: Secure DCI

6.5.30. AHB3 secure protection status register (RCU_AHB3SECP_STAT)

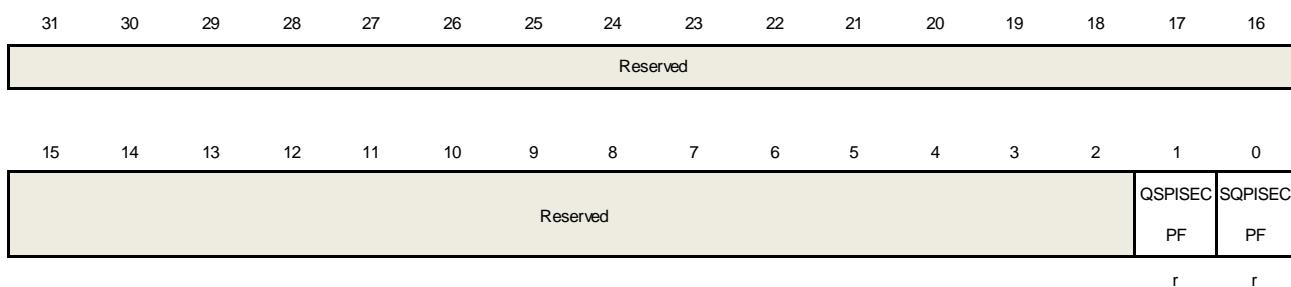
Address offset: 0xD0

Reset value: 0x0000 0000 (QSPI is not security aware)

Reset value: 0x0000 0002 (QSPI is security aware)

When TZEN = 1, this register provides AHB3 peripheral clock security status. Privileged and unprivileged, secure and non-secure accesses are all allowed access. When the peripheral is configured to be secure, the corresponding peripheral clock is also secure. When TZEN = 0, this register is RAZ

This register can 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	QSPISECPF	<p>QSPI security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure QSPI</p> <p>1: Secure QSPI</p>
0	SQPISECPF	<p>SQPI security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure SQPI</p> <p>1: Secure SQPI</p>

6.5.31. APB1 secure protection status register (RCU_APB1SECP_STAT)

Address offset: 0xD4

Reset value: 0x0000 0000

When TZEN = 1, this register provides APB1 peripheral clock security status. Privileged and unprivileged, secure and non-secure accesses are all allowed access. When the peripheral is configured to be secure, the corresponding peripheral clock is also secure. When TZEN = 0, this register is RAZ.

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	PMUSEC PF		Reserved		I2C1SEC PF	I2C0SEC PF		Reserved	USART0 SECPF	USART1 SECPF					
r					r	r			r	r		r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1SEC PF	Reserved	WWDGT SECPF		Reserved				TIMER5S ECPF	TIMER4S ECPF	TIMER3S ECPF	TIMER2S ECPF	TIMER1S ECPF		
r		r							r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	PMUSECPF	PMU security protection flag This flag is set by hardware when it is secure 0: Non secure PMU 1: Secure PMU
27:23	Reserved	Must be kept at reset value
22	I2C1SECPF	I2C1 security protection flag This flag is set by hardware when it is secure 0: Non secure I2C1 1: Secure I2C1
21	I2C0SECPF	I2C0 security protection flag This flag is set by hardware when it is secure 0: Non secure I2C0 1: Secure I2C0
20:19	Reserved	Must be kept at reset value
18	USART0SECPF	USART0 security protection flag This flag is set by hardware when it is secure 0: Non secure USART0 1: Secure USART0
17	USART1SECPF	USART1 security protection flag This flag is set by hardware when it is secure. 0: Non secure USART1 1: Secure USART1
16:15	Reserved	Must be kept at reset value
14	SPI1SECPF	SPI1 security protection flag This flag is set by hardware when it is secure 0: Non secure SPI1

		1: Secure SPI1
13:12	Reserved	Must be kept at reset value
11	WWDGTEC _{PF}	<p>WWDGTEC_{PF} security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure WWDGT</p> <p>1: Secure WWDGT</p>
10:5	Reserved	Must be kept at reset value
4	TIMER5SEC _{PF}	<p>TIMER5 security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure TIMER5</p> <p>1: Secure TIMER5</p>
3	TIMER4SEC _{PF}	<p>TIMER4 security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure TIMER4</p> <p>1: Secure TIMER4</p>
2	TIMER3SEC _{PF}	<p>TIMER3 security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure TIMER3</p> <p>1: Secure TIMER3</p>
1	TIMER2SEC _{PF}	<p>TIMER2 security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure TIMER2</p> <p>1: Secure TIMER2</p>
0	TIMER1SEC _{PF}	<p>TIMER1 security protection flag</p> <p>This flag is set by hardware when it is secure.</p> <p>0: Non secure TIMER</p> <p>1: Secure TIMER1</p>

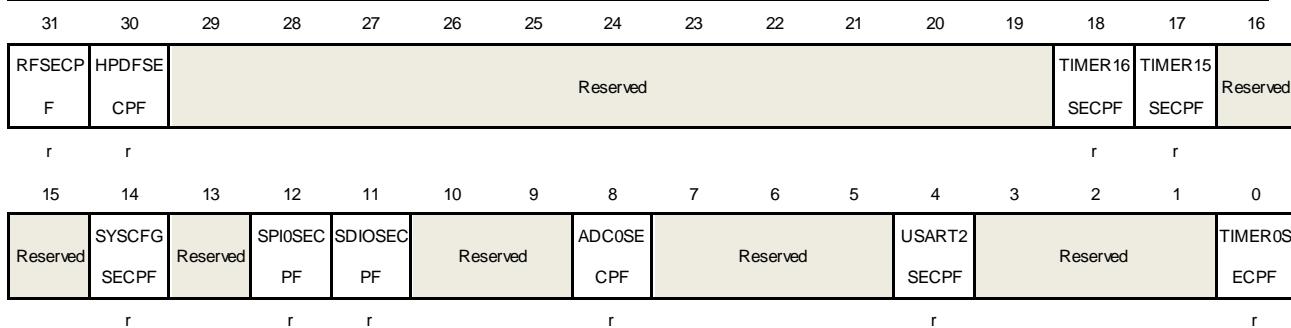
6.5.32. APB2 secure protection status register (RCU_APB2SECP_STAT)

Address offset: 0xD8

Reset value: 0x0000 0000

When TZEN = 1, this register provides APB2 peripheral clock security status. Privileged and unprivileged, secure and non-secure accesses are all allowed access. When the peripheral is configured to be secure, the corresponding peripheral clock is also secure. When TZEN = 0, this register is RAZ.

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



Bits	Fields	Descriptions
31	RFSECPF	RF security protection flag This flag is set by hardware when it is secure 0: Non secure RF 1: Secure RF
30	HPDFSECPF	HPDF security protection flag This flag is set by hardware when it is secure 0: Non secure HPDF 1: Secure HPDF
29:19	Reserved	Must be kept at reset value
18	TIMER16SEC _{PF}	TIMER16 security protection flag This flag is set by hardware when it is secure. 0: Non secure TIMER16 1: Secure TIMER16
17	TIMER15SEC _{PF}	TIMER15 security protection flag This flag is set by hardware when it is secure. 0: Non secure TIMER15 1: Secure TIMER15
16:15	Reserved	Must be kept at reset value
14	SYSCFGSEC _{PF}	SYSCFG security protection flag This flag is set by hardware when it is secure. 0: Non secure SYSCFG 1: Secure SYSCFG
13	Reserved	Must be kept at reset value
12	SPI0SEC _{PF}	SPI0 security protection flag This flag is set by hardware when it is secure. 0: Non secure SPI0 1: Secure SPI0
11	SDIOSEC _{PF}	SDIO security protection flag This flag is set by hardware when it is secure.

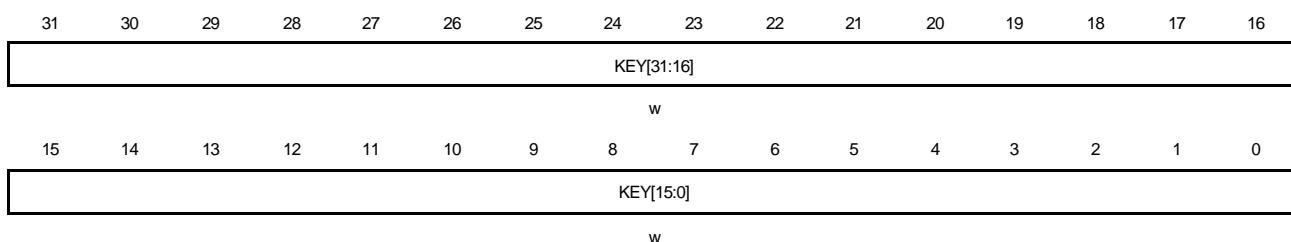
		0: Non secure SDIO 1: Secure SDIO
10:9	Reserved	Must be kept at reset value
8	ADC0SECPF	ADC0 security protection flag This flag is set by hardware when it is secure. 0: Non secure ADC0 1: Secure ADC0
7:5	Reserved	Must be kept at reset value
4	USART2SEC PF	USART2 security protection flag This flag is set by hardware when it is secure. 0: Non secure USART2 1: Secure USART2
3:1	Reserved	Must be kept at reset value
0	TIMER0SECPF	TIMER0 security protection flag This flag is set by hardware when it is secure. 0: Non secure TIMER0 1: Secure TIMER0

6.5.33. Voltage key register (RCU_VKEY)

Address offset: 0x100

Reset value: 0x0000 0000.

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



Bits	Fields	Descriptions
31:0	KEY[31:0]	The key of RCU_DSV register These bits are written only by software and read as 0. Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_DSV, register can be written.

6.5.34. Deep-sleep mode voltage register (RCU_DSV)

Address offset: 0x134

Reset value: 0x0000 0000

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

rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	DSLPVS[1:0]	Deep-sleep mode voltage select These bits are set and reset by software. 00 : The core voltage is 1.1V in Deep-sleep mode 01 : The core voltage is 1.0V in Deep-sleep mode 10 : The core voltage is 0.9V in Deep-sleep mode 11 : The core voltage is 0.8V in Deep-sleep mode

7. Interrupt/event controller (EXTI)

7.1. 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 controls power management. 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 29 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.

7.2. Characteristics

- Cortex-M33 system exception
- Up to 76 maskable peripheral interrupts for GD32W51x series
- 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 29 independent edge detectors in EXTI
- Three trigger types: rising, falling and both edges
- Software interrupt or event trigger
- Trigger sources configurable
- Secure events – The access to control and configuration bits of secure input events can be made secure and/or privileged

7.3. 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 following tables list all exception types.

Table 7-1. NVIC exception types in Cortex-M33

Exception Type	Vector Number	Priority (a)	Vector Address	Description
-	0	-	0x0000_0000	Reserved
Reset	1	-4	0x0000_0004	Reset
NMI	2	-2	0x0000_0008	Non maskable interrupt
HardFault	3	-3	0x0000_000C	Secure HardFault when AIRCR.BFHFNMIN is 1
		-1		Secure HardFault when AIRCR.BFHFNMIN is 0
		-1		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
SecureFault	7	Programmable	0x0000_001C	Secure fault
-	8-10	-	0x0000_0020 - 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

Note: When the processor is without the Security Extension, the priority of HardFault is -1 and vector address 0x0000_001C is reserved. When TrustZone is enabled, exception vector numbers 1~15 are banked or not between secure and non-secure, please refer to the corresponding kernel documentation for details.

The SysTick calibration value is 22500 and SysTick clock frequency is fixed to HCLK/8. So this will give a 1ms SysTick interrupt if HCLK is configured to 180MHz.

In an implementation with the Security Extension, each interrupt is configured by Secure software in Secure or Non-secure state, using NVIC_ITNS.

Table 7-2. Interrupt vector table

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 0	16	Window watchdog timer interrupt	0x0000_0040
IRQ 1	17	LVD through EXTI Line detection interrupt	0x0000_0044
IRQ 2	18	RTC Tamper andTimeStamp events interrupt	0x0000_0048

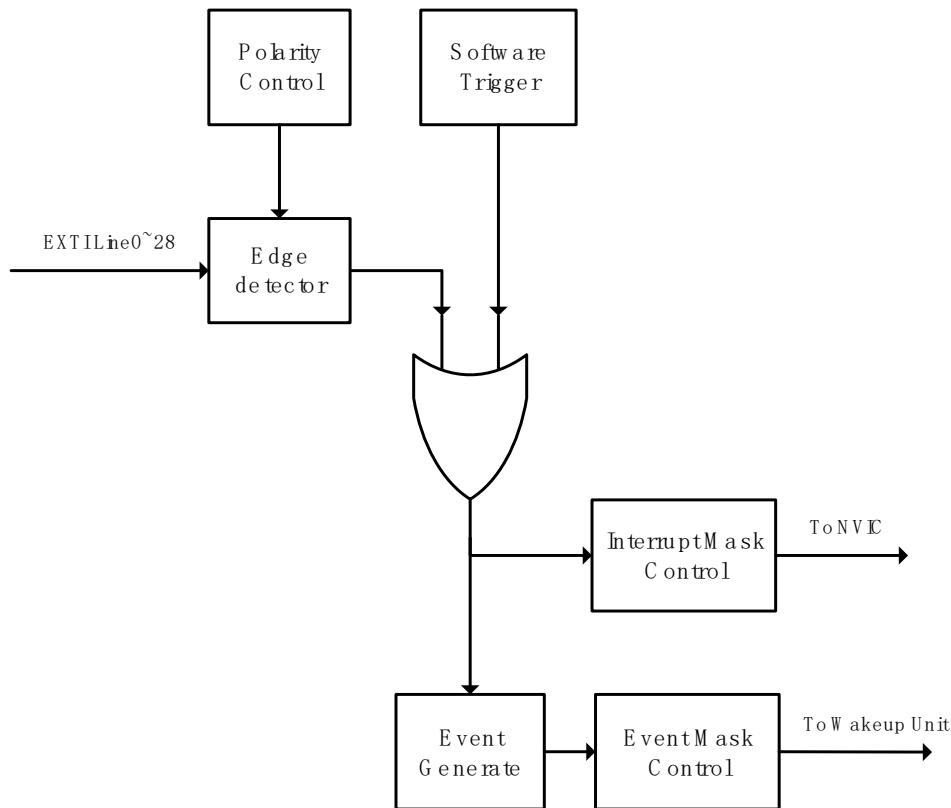
Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 3	19	RTC Wakeup event interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU global interrupt	0x0000_0054
IRQ 6	22	EXTI Line0 interrupt	0x0000_0058
IRQ 7	23	EXTI Line1 interrupt	0x0000_005C
IRQ 8	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	DMA0 channel6 global interrupt	0x0000_0084
IRQ 18	34	DMA0 channel7 global interrupt	0x0000_0088
IRQ 19	35	ADC interrupt	0x0000_008C
IRQ 20	36	RTC Tamper andTimeStamp events security interrupt	0x0000_0090
IRQ 21	37	RTC Wakeup event security interrupt	0x0000_0094
IRQ 22	38	RTC Alarm event security interrupt	0x0000_0098
IRQ 23	39	EXTI Line5-9 interrupt	0x0000_009C
IRQ 24	40	TIMER0 Break interrupt	0x0000_00A0
IRQ 25	41	TIMER0 update interrupt	0x0000_00A4
IRQ 26	42	TIMER0 commutation interrupt	0x0000_00A8
IRQ 27	43	TIMER0 Capture Compare interrupt	0x0000_00AC
IRQ 28	44	TIMER1 global interrupt	0x0000_00B0
IRQ 29	45	TIMER2 global interrupt	0x0000_00B4
IRQ 30	46	TIMER3 global interrupt	0x0000_00B8
IRQ 31	47	I2C0 event interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1/I2S1 global interrupt	0x0000_00D0
IRQ 37	53	USART0 global interrupt	0x0000_00D4
IRQ 38	54	USART1 global interrupt	0x0000_00D8
IRQ 39	55	USART2 global interrupt	0x0000_00DC
IRQ 40	56	EXTI Line10-15 interrupt	0x0000_00E0
IRQ 41	57	RTC Alarm event interrupt	0x0000_00E4

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 42	58	VLVDF interrupt	0x0000_00E8
IRQ 43	59	Reserved	0x0000_00EC
IRQ 44	60	TIMER15 global interrupt	0x0000_00F0
IRQ 45	61	TIMER16 global interrupt	0x0000_00F4
IRQ 46	62	Reserved	0x0000_00F8
IRQ 47	63	Reserved	0x0000_00FC
IRQ 48	64	Reserved	0x0000_0100
IRQ 49	65	SDIO global interrupt	0x0000_0104
IRQ50	66	TIMER4 global interrupt	0x0000_0108
IRQ51	67	I2C0 Wakeup interrupt	0x0000_010C
IRQ52	68	USART0 Wakeup	0x0000_0110
IRQ53	69	USART2 Wakeup	0x0000_0114
IRQ54	70	TIMER5 global interrupt	0x0000_0118
IRQ55	71	Reserved	0x0000_011C
IRQ56	72	DMA1 channel0 global interrupt	0x0000_0120
IRQ57	73	DMA1 channel1 global interrupt	0x0000_0124
IRQ58	74	DMA1 channel2 global interrupt	0x0000_0128
IRQ59	75	DMA1 channel3 global interrupt	0x0000_012C
IRQ60	76	DMA1 channel4 global interrupt	0x0000_0130
IRQ61	77	DMA1 channel5 global interrupt	0x0000_0134
IRQ62	78	DMA1 channel6 global interrupt	0x0000_0138
IRQ63	79	DMA1 channel7 global interrupt	0x0000_013C
IRQ64~65	80~81	Reserved	0x0000_0140-0x0000_0144
IRQ66	82	Wi-Fi11N wakeup interrupt	0x0000_0148
IRQ67	83	USBFS global interrupt	0x0000_014C
IRQ68	84	Reserved	0x0000_0150
IRQ69	85	Reserved	0x0000_0154
IRQ70	86	Reserved	0x0000_0158
IRQ71~75	87~91	Reserved	0x0000_015C-0x0000_016C
IRQ76	92	USBFS wakeup interrupt	0x0000_0170
IRQ77	93	Reserved	0x0000_0174
IRQ78	94	DCI global interrupt	0x0000_0178
IRQ79	95	CAU global interrupt	0x0000_017C
IRQ80	96	HAU/TRNG global interrupt	0x0000_0180
IRQ81	97	FPU global interrupt	0x0000_0184
IRQ82~88	98~104	Reserved	0x0000_0188-0x0000_01A0
RQ89	105	HPDF global interrupt0	0x0000_01A4

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
RQ90	106	HPDF global interrupt1	0x0000_01A8
IRQ91	107	Wi-Fi11N global interrupt0	0x0000_01AC
IRQ92	108	Wi-Fi11N global interrupt1	0x0000_01B0
IRQ93	109	Wi-Fi11N global interrupt2	0x0000_01B4
IRQ94	110	EFUSE global interrupt	0x0000_01B8
IRQ95	111	QSPI global interrupt	0x0000_01BC
IRQ96	112	PKCAU global interrupt	0x0000_01C0
IRQ97	113	TSI global interrupt	0x0000_01C4
IRQ98	114	ICACHE global interrupt	0x0000_01C8
IRQ99	115	TZIAC security interrupt	0x0000_01CC
IRQ100	116	FMC secure interrupt	0x0000_01D0
IRQ101	117	QSPI security interrupt	0x0000_01D4

7.4. External interrupt and event(EXTI) block diagram

Figure 7-1. Block diagram of EXTI



7.5. External interrupt and Eventfunction overview

The EXTI contains up to 29 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 13 lines from internal modules (including LVD, RTC, USART, USBFS, I2C and Wi-Fi). All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers in SYSCFG module (please refer to [System configuration controller \(SYSCFG\)](#) 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 events. EXTI can be used to wake up processor and the whole system when some expected event occurs, such as a special GPIO pin toggling or RTC alarm.

The internal trigger source from USART is able to generate event for waking up the system. However, the module is also requested to generate a synchronous interrupt for the processor to wake up the CPU from Deep-sleep mode. Both internal trigger lines can be masked by setting corresponding INTEN and EVEN registers in EXTI module.

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 SYSCFG 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_EVEN bits.
4. EXTI starts to detect changes on the configured pins. The related PDx bits in EXTI_PD will be set when desired change is detected on these pins and thus, trigger interrupt or event for software. 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_EVEN bits.
2. Set SWIEVx bits in EXTI_SWIEV register. The related PDx bits will be set immediately and thus, trigger interrupts or events. Software should response to these interrupts, and clear related PDx bits.

Table 7-3. EXTI source

EXTI Line Number	Source
0	PA0 / PB0 / PC0

EXTI Line Number	Source
1	PA1 / PB1 / PC1
2	PA2 / PB2 / PC2
3	PA3 / PB3 / PC3
4	PA4 / PB4 / PC4
5	PA5 / PB5 / PC5
6	PA6 / PB6 / PC6
7	PA7 / PB7 / PC7
8	PA8 / PB8 / PC8
9	PA9 / PB9
10	PA10 / PB10
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC Alarm non-secure
18	RTC Alarm secure
19	VLVDF
20	Wi-Fi11N wakeup
21	USBFS wakeup
22	RTC Tamper and Timestamp non-secure
23	RTC Tamper and Timestamp secure
24	RTC Wakeup event non-secure
25	RTC Wakeup event secure
26	I2C0 Wakeup event
27	USART0 Wakeup
28	USART2 Wakeup

7.6. EXTI event protection

The EXTI is able to protect event register bits from being modified by non-secure and unprivileged accesses. The protection can individually be activated per input event via the register bits in EXTI_SECCFG and EXTI_PRIVCFG. At EXTI level the protection consists in preventing unauthorized write access to:

- Change the settings of the secure and/or privileged configurable events.
- Change the masking of the secure and/or privileged input events.
- Clear pending status of the secure and/or privileged input events.

Table 7-4. Register protection overview

Register name	Access type	Protection ⁽¹⁾⁽²⁾
EXTI_INTEN	RW	Security and privilege can be bit wise enabled in EXTI_SECCFG and EXTI_PRIVCFG.
EXTI_EVEN	RW	
EXTI_RTEN	RW	
EXTI_FTEN	RW	
EXTI_SWIEV	RW	
EXTI_PD	RW	
EXTI_SECCFG	RW	Always secure, and privilege can be bit wise enabled in EXTI_PRIVCFG.
EXTI_PRIVCFG	RW	Always privilege, and security can be bit wise enabled in EXTI_SECCFG.
EXTI_LOCK	RW	Always secure.

1. Security is enabled with the individual input event (EXTI_SECCFG register).

2. Privilege is enabled with the individual input event (EXTI_PRIVCFG register).

7.7. EXTI security protection

When security is enabled for an input event, the associated input event configuration and control bits can only be modified and read by a secure access, a non-secure write access is discarded and a read returns 0.

When input events are non-secure, the security is disabled. The associated input event configuration and control bits can be modified and read by a secure access and non-secure access. The security configuration in registers EXTI_SECCFG can be globally locked after reset by EXTI_LOCK.LOCK.

7.8. EXTI privilege protection

When privilege is enabled for an input event, the associated input event configuration and control bits can only be modified and read by a privilege access, an unprivileged write access is discarded and a read returns 0.

When input events are unprivileged, the privilege is disabled. The associated input event configuration and control bits can be modified and read by a privilege access and unprivileged access. The privilege configuration in registers EXTI_PRIVCFG can be globally locked after reset by EXTI_LOCK.LOCK.

7.9. Register definition

EXTI secure access base address: 0x5001 3C00
 EXTI non-secure access base address: 0x4001 3C00

7.9.1. Interrupt enable register (EXTI_INTEN)

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			INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
			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
31:29	Reserved	Must be kept at reset value
28: 0	INTENx	<p>Interrupt enable bit x(x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, INTENx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, INTENx can only be accessed with secure access. Non-secure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, INTENx can be accessed with unprivileged and privilege access.</p> <p>When EXTI_PRIVCFG PRIVx is enabled, INTENx can only be accessed with privilege access. Unprivileged write to this bit x is discarded, unprivileged read returns 0.</p> <p>0: Interrupt from Linex is disabled</p> <p>1: Interrupt from Linex is enabled</p>

7.9.2. Event enable register (EXTI_EVENT)

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			EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
			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
31:29	Reserved	Must be kept at reset value
28: 0	EVENx	<p>Event enable bit x(x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, EVENx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, EVENx can only be accessed with secure access. Nonsecure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, EVENx can be accessed with unprivileged and privilege access.</p> <p>When EXTI_PRIVCFG PRIVx is enabled, EVENx can only be accessed with privilege access.Unprivileged write to this bit x is discarded, unprivileged read returns 0.</p> <p>0: Event from Linex is disabled</p> <p>1: Event from Linex is enabled</p>

7.9.3. 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		RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16	
		rw	rw												
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
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:0	RTENx	<p>Rising edge trigger enable (x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, RTENx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, RTENx can only be accessed with secure access. Nonsecure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, RTENx can be accessed with unprivileged and privilege access.</p> <p>When EXTI_PRIVCFG PRIVx is enabled, RTENx can only be accessed with privilege access.</p>

privilege access.Unprivileged write to this bit x is discarded, unprivileged read returns 0.

0: Rising edge of Linex is invalid

1: Rising edge of Linex is valid as an interrupt/event request

7.9.4. Falling edge trigger enable register (EXTI_FTEN)

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		FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16	
		rw	rw												
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
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31: 29	Reserved	Must be kept at reset value
28: 0	FTENx	<p>Falling edge trigger enable (x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, FTENx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, FTENx can only be accessed with secure access. Nonsecure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, FTENx can be accessed with unprivileged and privilege access.</p> <p>When EXTI_PRIVCFG PRIVx is enabled, FTENx can only be accessed with privilege access.Unprivileged write to this bit x is discarded, unprivileged read returns 0.</p> <p>0: Falling edge of Linex is invalid</p> <p>1: Falling edge of Linex is valid as an interrupt/event request</p>

7.9.5. 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		SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16	
		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
31:29	Reserved	Must be kept at reset value
28: 0	SWIEVx	<p>Interrupt/Event softw are trigger (x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, SWIEVx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, SWIEVx can only be accessed with secure access. Nonsecure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, SWIEVx can be accessed with unprivileged and privilege access.</p> <p>When EXTI_PRIVCFG PRIVx is enabled, SWIEVx can only be accessed with privilege access. Unprivileged write to this bit x is discarded, unprivileged read returns 0.</p> <p>0: Deactivate the EXTIx software interrupt/event request</p> <p>1: Activate the EXTIx software interrupt/event request</p>

7.9.6. Pending register (EXTI_PD)

Address offset: 0x14

Reset value: undefined

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	PD28	PD27	PD26	PD25	PD24	PD23	PD22	PD21	PD20	PD19	PD18	PD17	PD16		
	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: 29	Reserved	Must be kept at reset value
28: 0	PDx	<p>Interrupt pending status (x=0..28)</p> <p>When EXTI_SECCFG SECx is disabled, PDx can be accessed with non-secure and secure access.</p> <p>When EXTI_SECCFG SECx is enabled, PDx can only be accessed with secure access. Nonsecure write to this bit x is discarded, non-secure read returns 0.</p> <p>When EXTI_PRIVCFG PRIVx is disabled, PDx can be accessed with unprivileged and privilege access.</p>

When EXTI_PRIVCFG PRIVx is enabled, PDx can only be accessed with privilege access. Unprivileged write to this bit x is discarded, unprivileged read returns 0.

0: EXTI Linex is not triggered

1: EXTI Linex is triggered. This bit is cleared to 0 by writing 1 to it.

7.9.7. Security configuration register (EXTI_SECCFG)

Address offset: 0x18

System reset: 0x0000 0000

This register provides write access security, a non-secure write access is ignored and causes the generation of an illegal access event. A non-secure read returns the register data.

Contains only register bits for security capable input events.

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	SEC28	SEC27	SEC26	SEC25	SEC24	SEC23	SEC22	SEC21	SEC20	SEC19	SEC18	SEC17	SEC16
			rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:0	SECx	<p>Security enable on event input x (where x = 0 to 28)</p> <p>When EXTI_PRIVCFG.PRIVx is disabled, SECx can be accessed with privilege and unprivileged access.</p> <p>When EXTI_PRIVCFG.PRIVx is enabled, SECx can only be written with privilege access. Unprivileged write to this SECx is discarded.</p> <p>0: Event security disabled (non-secure)</p> <p>1: Event security enabled (secure)</p>

7.9.8. Privilege configuration register (EXTI_PRIVCFG)

Address offset: 0x1C

System reset: 0x0000 0000

This register provides privileged write access protection, an unprivileged write access is discarded and causes the generation of an illegal access event. An unprivileged read returns the register data.

Contains only register bits for privilege capable input events.

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	PRIV28	PRIV27	PRIV26	PRIV25	PRIV24	PRIV23	PRIV22	PRIV21	PRIV20	PRIV19	PRIV18	PRIV17	PRIV16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV15	PRIV14	PRIV13	PRIV12	PRIV11	PRIV10	PRIV9	PRIV8	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:0	PRIVx	<p>Privilege enable on event input x (where x = 0 to 28)</p> <p>When EXT1_SECCFG.SECx is disabled, PRIVx can be accessed with secure and nonsecure access.</p> <p>When EXT1_SECCFG.SECx is enabled, PRIVx can only be written with secure access. Non-secure write to this PRIVx is discarded.</p> <p>0: Event privilege disabled (unprivileged)</p> <p>1: Event privilege enabled (privileged)</p>

7.9.9. Lock register (EXTI_LOCK)

Address offset: 0x20

System reset: 0x0000 0000

This register provides both write access security, a non-secure write access is ignored and a read access returns zero data, and generate an illegal access event.

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

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	LOCK	<p>Global security and privilege configuration registers EXTL_SECCFG and EXTL_PRIVCFG lock.</p> <p>This register's bit is written once after reset.</p> <p>0: Security and privilege configuration open, can be modified.</p> <p>1: Security and privilege configuration locked, can no longer be modified.</p>

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

8.1. Overview

There are up to 43 general purpose I/O pins (GPIO), named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC8, and PC14 ~ PC15 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.

8.2. 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

8.3. Function overview

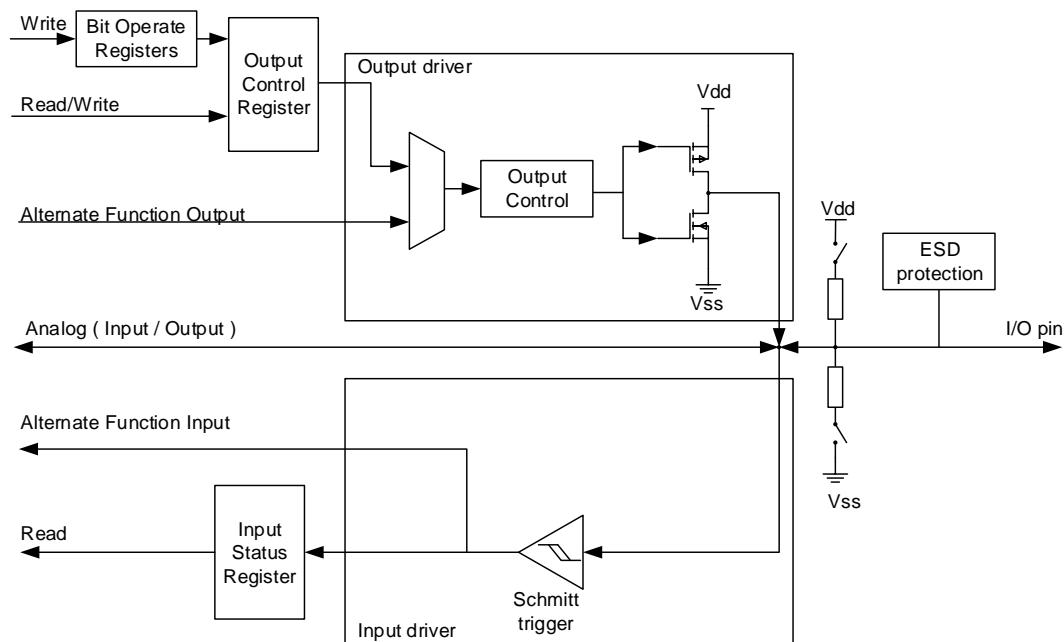
Each of the general-purpose I/O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output 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 8-1. 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 8-1. Basic structure of a standard I/O port bit shows the basic structure of an I/O Port bit.

Figure 8-1. Basic structure of a standard I/O port bit


8.3.1. 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. But the JTAG/Serial-Wired Debug pins are in input PU/PD mode after reset:

- PA15: JTDI in PU mode
- PA14: JTCK / SWCLK in PD mode
- PA13: JTMS / SWDIO in PU mode
- PB4: NJTRST in PU mode
- PB3: JTDI in Floating mode

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 (GPIO_x_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 (GPIO_x_OCTL) is output on the I/O pin.

There is no need to read-then-write when programming the GPIO_x_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 (GPIO_x_BOP, or for clearing only GPIO_x_BC, or for toggle only GPIO_x_TG). The other bits will not be affected.

8.3.2. External interrupt/event lines

All ports have external interrupt capability. To use external interrupt lines, the port must be configured as input mode.

8.3.3. 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_AFSELz ($z = 0, 1$)). The detail alternate function assignments for each port are in the device datasheet.

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

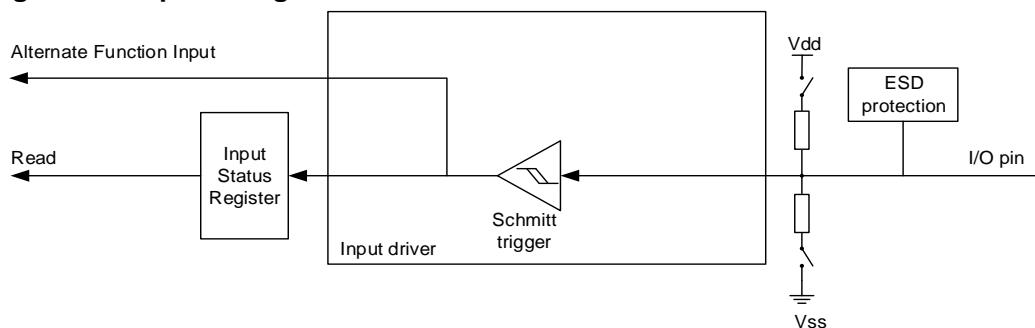
8.3.5. Input configuration

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.
- The output buffer is disabled.

[Figure 8-2. Input configuration](#) shows the input configuration.

Figure 8-2. Input configuration



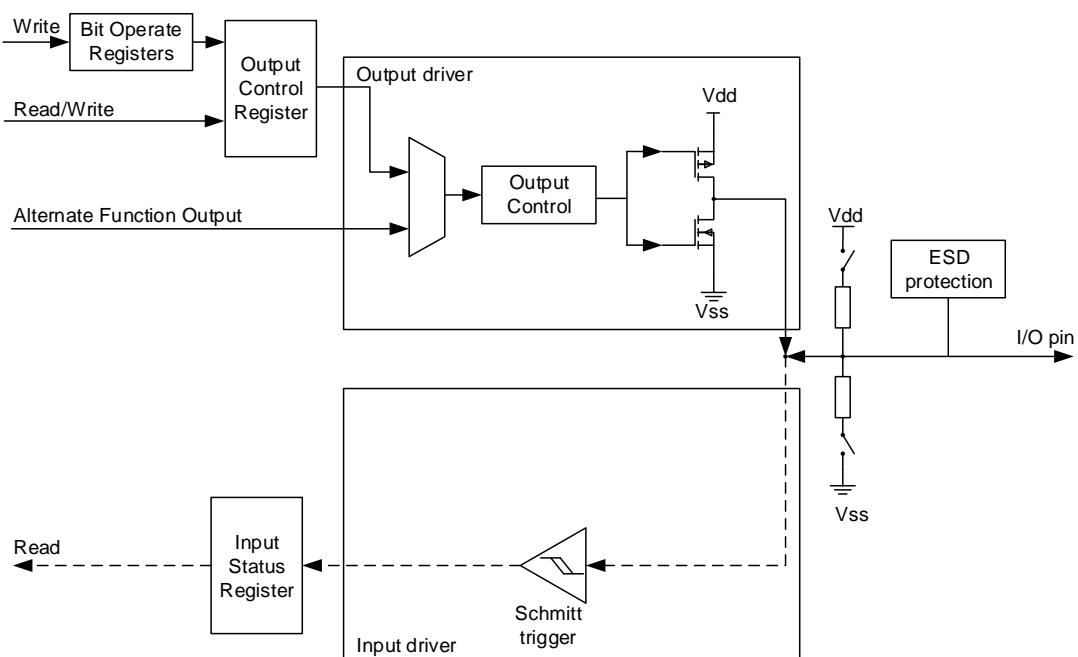
8.3.6. Output configuration

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 8-3. Output configuration**](#) shows the output configuration.

Figure 8-3. Output configuration



8.3.7. 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 8-4. Analog configuration**](#) shows the analog configuration.

Figure 8-4. Analog configuration



8.3.8. Alternate function (AF) configuration

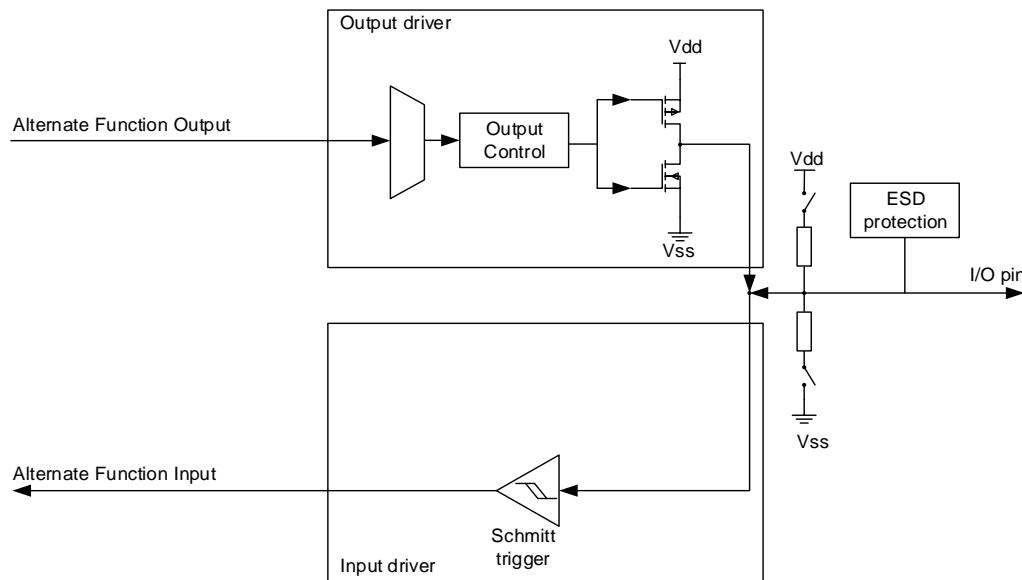
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

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 8-5. Alternate function configuration**](#) shows the alternate function configuration.

Figure 8-5. Alternate function configuration



8.3.9. 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 and

GPIO_x_AFSEL_z ($z=0, 1$). It allows the I/O configuration to be frozen by the 32-bit locking register (GPIO_x_LOCK). When the special LOCK sequence has occurred on LKK bit in GPIO_x_LOCK register and the LKy bit is set in GPIO_x_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.

8.3.10. 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 GPIO_x_TG register. The output signal frequency could up to the half of the AHB clock.

8.4. TrustZone GPIO security

If TZEN=1, after reset, all GPIO ports are secure, secure code can use GPIO_x_SCFG register to configure GPIO pin secure state, once one pin set to secure its corresponding configuration bits for alternate function, control bits, mode selection, reset/set bits, lock bits, I/O data, are secure. [Table 8-2. GPIO secure state](#) show GPIO secure state.

Table 8-2. GPIO secure state

GPIO _x _SCFG	Register	Bit/Bits	Secure state	
SCFG _y = 1	GPIO _x _CTL	CTLy[1:0]	Non-secure read is zero, write will ignore	
	GPIO _x _OMODE	OMy		
	GPIO _x _OSPD	OSPDy[1:0]		
	GPIO _x _PUD	PUDy[1:0]		
	GPIO _x _ISTAT	ISTATy		
	GPIO _x _OCTL	OCTLy		
	GPIO _x _BOP	BOPy/CRy		
	GPIO _x _LOCK	LKy		
	GPIO _x _AFSEL0	SELy[3:0]		
	GPIO _x _AFSEL1			
	GPIO _x _BC	CRy		
	GPIO _x _TG	TGy		

Note: GPIO_x, $x=A\ldots C$, and $y=0..15$

If one pin of GPIO is used for peripheral, whether the pin can be used is determined with the secure state of GPIO pin and peripheral. If peripheral is secure but GPIO pin is non-secure then the pin can not use, read/write is all zero; if peripheral is non-secure but GPIO pin is secure then the pin can use; for analog peripheral like ADC, the secure state must to be same or it will block between peripheral and GPIO pin.

8.5. Register definition

GPIOA secure access base address: 0x5002 0000

GPIOA non-secure base address: 0x4002 0000

GPIOB secure access base address: 0x5002 0400

GPIOB non-secure base address: 0x4002 0400

GPIOC secure access base address: 0x5002 0800

GPIOC non-secure base address: 0x4002 0800

8.5.1. Port control register (GPIOx_CTL, x=A.C)

Address offset: 0x00

Reset value: The reset value is determined by the FW AES Key(Firmware AES Key) bit0 and bit1 in the EFUSE. When the bit0 is 1, the PA4/PA5/PA6/PA7/PB3/PB4 is configured as one set of QSPI port automatically by the hardware, and when the bit1 is 1, the PA9/PA10/PA11/PA12/PC4/PC5 is configured as the other set of QSPI port automatically by the hardware as well. FW AES Key[1:0] default value is 00. Please refer to the table [Table 8-3. GPIOx_CTL reset value](#) below for this register reset value.

Table 8-3. GPIOx_CTL reset value

FW AES Key[1:0]	GPIOA_CTL	GPIOB_CTL	GPIOC_CTL
00	0xA800 0000	0x0000 0280	0x0000 0000
01	0xA800 AA00	0x0000 0280	0x0000 0000
10	0xAAA8 0000	0x0000 0280	0x0000 0A00
11	0xAAA8 AA00	0x0000 0280	0x0000 0A00

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

8.5.2. Port output mode register (GPIOx_OMODE, x=A..C)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)/half-word(16-bit)/byte(8-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
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0

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

8.5.3. Port output speed register (GPIOx_OSPD, x=A.C)

Address offset: 0x08

Reset value: The reset value is determined by the FW AES Key bit0 and bit1 in the EFUSE.

When the bit0 is 1, the PA4/PA5/PA6/PA7/PB3/PB4 is configured as one set of QSPI port automatically by the hardware, and when the bit1 is 1, the PA9/PA10/PA11/PA12/PC4/PC5 is configured as the other set of QSPI port automatically by the hardware as well. FW AES Key[1:0] default value is 00. Please refer to the table [Table 8-4. GPIOx_OSPD reset value](#) below for this register reset value.

Table 8-4. GPIOx_OSPD reset value

FW AES Key[1:0]	GPIOA_OSPD	GPIOB_OSPD	GPIOC_OSPD
00	0x0C00 0000	0x0000 00C0	0x0000 0000
01	0x0C00 FF00	0x0000 03C0	0x0000 0000
10	0x0FFC 0000	0x0000 00C0	0x0000 0F00
11	0x0FFC FF00	0x0000 03C0	0x0000 0F00

This register has to be accessed by word(32-bit)/half-word(16-bit)/byte(8-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	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	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. 00: Output speed level 0 (reset state) 01: Output speed level 1 10: Output speed level 2 11: Output speed level 3

8.5.4. Port pull-up/pull-down register (GPIOx_PUD, x=A.C)

Address offset: 0x0C

Reset value: The reset value is determined by the FW AES Key bit0 and bit1 in the EFUSE.

When the bit0 is 1, the PA4/PA5/PA6/PA7/PB3/PB4 is configured as one set of QSPI port automatically by the hardware, and when the bit1 is 1, the PA9/PA10/PA11/PA12/PC4/PC5 is configured as the other set of QSPI port automatically by the hardware as well. FW AES Key[1:0] default value is 00. Please refer to the table [Table 8-5. GPIOx_PUD reset value](#) below for this register reset value.

Table 8-5. GPIOx_PUD reset value

FW AES Key[1:0]	GPIOA_PUD	GPIOB_PUD	GPIOC_PUD
00	0x6400 0000	0x0000 0100	0x0000 0000
01	0x6400 A000	0x0000 0280	0x0000 0000
10	0x6680 0000	0x0000 0100	0x0000 0A00
11	0x6680 A000	0x0000 0280	0x0000 0A00

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

8.5.5. Port input status register (GPIOx_ISTAT, x=A..C)

Address offset: 0x10

Reset value: 0x0000 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0

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

8.5.6. Port output control register (GPIOx_OCTL, x=A..C)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)/half-word(16-bit)/byte(8-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
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0

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

8.5.7. Port bit operate register (GPIOx_BOP, x=A..C)

Address offset: 0x18

Reset value: 0x0000 0000

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

8.5.8. Port configuration lock register (GPIOx_LOCK, x=A..C)

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
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"

8.5.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A.C)

Address offset: 0x20

Reset value: The reset value is determined by the FW AES Key bit0 and bit1 in the EFUSE. When the bit0 is 1, the PA4/PA5/PA6/PA7/PB3/PB4 is configured as one set of QSPI port automatically by the hardware, and when the bit1 is 1, the PA9/PA10/PA11/PA12/PC4/PC5 is configured as the other set of QSPI port automatically by the hardware as well. FW AES Key[1:0] default value is 00. Please refer to the table [Table 8-6. GPIOx_AFSEL0 reset value](#) below for this register reset value.

Table 8-6. GPIOx_AFSEL0 reset value

FW AES Key[1:0]	GPIOA_AFSEL0	GPIOB_AFSEL0	GPIOC_AFSEL0
00	0x0000 0000	0x0000 0000	0x0000 0000
01	0x3333 0000	0x0003 3000	0x0000 0000
10	0x0000 0000	0x0000 0000	0x0033 0000
11	0x3333 0000	0x0003 3000	0x0033 0000

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

8.5.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A.C)

Address offset: 0x24

Reset value: The reset value is determined by the FW AES Key bit0 and bit1 in the EFUSE. When the bit0 is 1, the PA4/PA5/PA6/PA7/PB3/PB4 is configured as one set of QSPI port automatically by the hardware, and when the bit1 is 1, the PA9/PA10/PA11/PA12/PC4/PC5 is configured as the other set of QSPI port automatically by the hardware as well. FW AES Key[1:0] default value is 00. Please refer to the table [Table 8-7. GPIOx_AFSEL1 reset value](#) below for this register reset value.

Table 8-7. GPIOx_AFSEL1 reset value

FW AES Key[1:0]	GPIOA_AFSEL1	GPIOB_AFSEL1	GPIOC_AFSEL1
00	0x0000 0000	0x0000 0000	0x0000 0000

01	0x0000 0000	0x0000 0000	0x0000 0000
10	0x0004 4440	0x0000 0000	0x0000 0000
11	0x0004 4440	0x0000 0000	0x0000 0000

This register has to be accessed by word(32-bit)/half-word(16-bit)/byte(8-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]			
15	14	13	12	11	10	8	7	6	5	4	3	2	1	0	
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			

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

8.5.11. Bit clear register (GPIOx_BC, x=A.C)

Address offset: 0x28

Reset value: 0x0000 0000

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

8.5.12. Port bit toggle register (GPIOx_TG, x=A.C)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)/half-word(16-bit)/byte(8-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
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
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	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

8.5.13. GPIO secure configuration register (GPIOx_SCFG) (x=A...C)

Address offset: 0x30

Reset value: 0x0000 FFFF

If TZEN = 0 the register is RAZ/WI, if TZEN = 1, secure code can use this register to configure GPIO pin secure state, non-secure code can read, but write will ignore.

This register has to be accessed by word (32-bit)/half-word (16-bit)/byte (8-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
SCFG15	SCFG14	SCFG13	SCFG12	SCFG11	SCFG10	SCFG9	SCFG8	SCFG7	SCFG6	SCFG5	SCFG4	SCFG3	SCFG2	SCFG1	SCFG0
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	SCFGy(y=0~15)	<p>GPIOx secure configure bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure the I/O pin to non-secure</p> <p>1: Configure the I/O pin to secure.</p>

[**Table 8-2. GPIO secure state**](#) show GPIO secure state.

9. TrustZone protection controller union (TZPCU)

9.1. Overview

This section describes the TrustZone® protection controller union. Three different sub-blocks, TrustZone® security privilege controller (TZSPC), TrustZone® block-based memory protection controller (TZBMP) and TrustZone® illegal access controller (TzIAC), are used to configure system security or privilege in a product with programmable-security and privileged attributes. TZSPC is used to define the secure/privilege state for securable slave/master peripherals. TrustZone® mark memory protection controller (TZMMPC) do the security checking of off-chip memories based on the size of non-secure area which is defined in TZSPC. For the on-chip RAM, the security checking is done based on block level which is configured by the TZBMP through an AHB interface. TzIAC is used to enable all illegal access events for slave/master peripherals in system. If an interrupt is enabled, a dedicated interrupt signal is asserted and generates a secure interrupt towards NVIC whenever a security violation is detected. The interrupt is cleared by writing to the appropriate register of TzIAC.

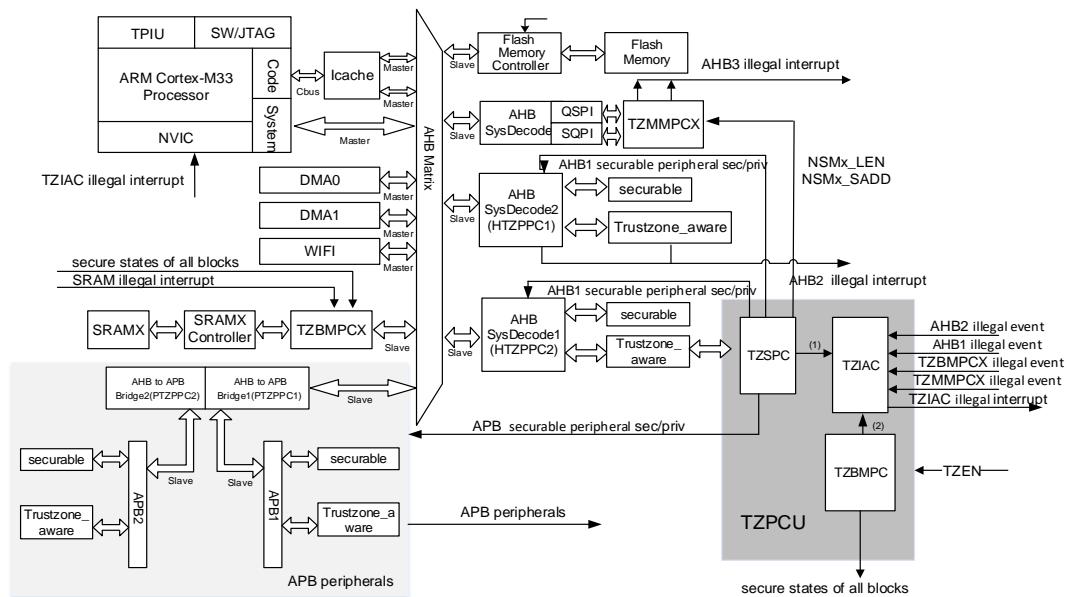
9.2. Characteristics

- TZSPC, TZBMP and TzIAC have independent 32-bit AHB interface.
- For TZSPC, whether non-secure/non-privilege access is supported is defined by secure/privilege configuration registers.
- For TZBMP and TzIAC, only secure access is supported.
- For securable slave/master peripherals, secure/privilege state is defined in TZSPC registers.
- For off-chip memories, the size of non-secure area is defined in TZSPC registers.
- For on-chip RAM, the secure states of all blocks is defined in TZBMP registers.

9.3. Function overview

9.3.1. Block diagram

Figure 9-1. Block diagram of TZPCU



There are three different sub-blocks, TrustZone® security privilege controller (TZSPC), TrustZone® block-based memory protection controller (TZBMPC) and TrustZone® illegal access controller (TZIAC) in TZPCU. These are the union function of TrustZone® protection controller which is beyond AHB and ARMv8-M. These beyond functions are realized through: APB TrustZone® peripheral protection controller (PTZPPC) in AHB/APB bridge gates transactions to, and responses from securable APB peripherals when a security violation occurs; AHB TrustZone® peripheral protection controller (HTZPPC) in AHB address decode gates transactions to, and responses from securable/privilege APB peripherals when a security violation occurs; TrustZone® block-based TZBMPCx firewalls gates transactions to, and responses from SRAMs on chip; TrustZone® mark TZMMPCx firewalls gates transactions to, and responses from memories off chip.

Figure 9-1. Block diagram of TZPCU shows the ARMv8-M (Cortex-M33) security architecture with secure, securable and TrustZone-aware peripherals, these peripherals are divided by PTZPPC and HTZPPC, and introduced by [Table 9-1. TrustZone® peripherals](#).

Table 9-1. TrustZone® peripherals

Peripherals	Introduce
privilege	TZSPC define whether the peripherals can connect to PTZPPC/HTZPPC
securable	TZSPC define whether the peripherals can connect to PTZPPC/HTZPPC
secure	Peripherals are always secure connect to PTZPPC/ HTZPPC (such as TZIAC and TZBMPC)
non-secure and non-privilege	Peripherals do not filter with PTZPPC/HTZPPC, connected directly to AHB/APB
TrustZone-aware	Refer to TrustZone-aware peripherals describe below .

1. According to security mode TrustZone-aware AHB masters can drive HNONSEC signal, secure transactions are signaled with HNONSEC = 0 on AHB bus, non-secure transactions are signaled with HNONSEC = 1 on AHB bus. Peripherals connected directly to AHB or APB bus and implementing a specific TrustZone® behavior (such as TZSPC, whether non- privilege access is supported for secure configuration registers is defined by privilege configuration registers).

9.3.2. Illegal access definition

The types of illegal access of peripheral register and memory (internal block-based SRAM or marked external memory) are different, as described below.

Peripheral register

- Illegal non-secure access

Non-secure transfers to secure peripheral are blocked and thus the addressed resource returns all zero data for read, ignores any write and generates an illegal access event for illegal access, but no bus error is generated. However transactions to secure and privilege configuration registers may have some differences, they are TrustZone-aware peripherals (for more information please refer to the security protection description of TrustZone-aware peripherals).

- Illegal non-privilege access

If a privilege resource is accessed by a non-privilege resource, this will be considered as illegal, but for this illegal access there is no event or bus error will generate, read access will return zero, write access will be ignored.

Memory (internal block-based SRAM or marked external memory)

- Illegal non-secure access

Non-secure transfers to secure block/regions are blocked and thus the addressed resource generates an illegal access event for illegal access, and a bus error for illegal fetch access.

- Illegal secure access

For SRAM or marked external memory, secure transfers to non-secure SRAM or external memory is illegal and a bus error will generate. For SRAM, when configure SRWACFG bit in the TZPCU_TZBMPCx_CTL register, it will make secure read/write aceess is legal, but secure execution aceess is still illegal.

- Illegal non-privilege access

Non-privilege transfers to privilege peripheral are blocked and thus the addressed resource returning all zero data for read, ignoring any write and do not generate an illegal access event and bus error.

NOTE: If during a burst, any of the beats of the burst tries to transfer to/from an illegal address, the module masks the rest of the burst, and forwards IDLE transfers on the master port.

9.3.3. TrustZone® security privilege controller (TZSPC)

The TZSPC is composed of a configurable set of registers defines which peripheral is secured and/or privileged, and it also controls the non-secure area size for the mark memory peripheral controller (MMPc).

Peripherals secure and privilege state are configured through:

- TZPCU_TZSPC_SAM_CFGx registers to control PTZPPC/HTZPPC firewall stubs for the securable peripherals.
- TZPCU_TZSPC_PAM_CFGx registers to control PTZPPC/HTZPPC firewall stubs for the privileged peripherals.
- TZPCU_TZSPC_TZMMPCx_NSMy registers to control TZMMPCx firewall stubs for the securable mark external memory.
- TZPCU_TZSPC_DBG_CFG registers to control MCU debug configuration.

The privilege configuration bit of given peripheral can be modified by a secure-privilege transaction when the peripheral has been configured as secure, otherwise a privileged transaction (non-secure) is sufficient. The definition of these privilege attributes is available even when TZEN = 0.

The secure configuration bit of given peripheral can be modified only with a secure-privilege transaction if the peripheral has been configured as privilege, otherwise a secure transaction (non-privileged) is sufficient.

For external memories SQPI_PSRAM (128MB) or QSPI_FLASH (128MB), they are marked memory, when TZEN=1, after reset they are all secure, the non-secure area is defined through:

- The TZPCU_TZSPC_TZMMPCx_NSMy register NSMy_SADD[13:0] defines the start position of the y-h non-secure area on external memory;
- The TZPCU_TZSPC_TZMMPCx_NSMy register NSMy_LEN[14:0] defines the length of the y-th non-secure area on external memory;

Note: x represents which kind external memory interface, y represents number of regions of the target external memory interface (such as SQPI_PSRAM). The total area considered as non-secure is the sum of the independent ones. An overlap of one section over the other one has no specific effect.

[Table 9-2. TZMMPCx](#) describes the characteristics of the available TZMMPCx

Table 9-2. TZMMPCx

MPC	Region	Number of regions(y)	Kind of memory interface
TZMMPC0 (extern memory 0)	Non secure mark	4 (0~3)	QSPI_flash

TZMMPC1 (extern memory 1)	Non secure mark	2 (0~1)	SQPI_PSRAM
---------------------------	-----------------	---------	------------

The TZSPC is a TrustZone-aware peripheral, meaning that secure and non-secure registers co-exist within the peripheral. An exception exists for the TZPCU_TZSPC_SAM_CFG and TZPCU_TZSPC_PAM_CFG: any read access, secure or not, are supported. A dedicated interrupt signal is asserted whenever a security violation is detected. The interrupt must be enabled by TZIAC. The interrupt is cleared by writing to the appropriate register.

9.3.4. TrustZone® block based memory protection controller (TZBMPc)

The TZBMPc gates transactions to the AHB master interface when a security violation occurs. TZBMPc is secure peripherals, thus systematically generate an illegal access event when accessed by a non-secure access.

For block-based memory protection controller (internal SRAM0~3), the non-secure and secure blocks are defined through:

- In TZPCU_TZBMPCx_VECy every bit defines one block (256 byte is see one block) secure state.
- In TZPCU_TZBMPCx_LOCK0 every bit locks a union block (32 blocks).

Note: y represents the number of union-blocks (union-block = 32 blocks of SRAM base block size). On the devices, if the SRAM size is 64 Kbytes and the block size is 256 bytes, then union-block size is $32 * 256 = 8$ Kbytes and $64 / 8 - 1$ is 7 means y is 0~7. It means 8 vector registers (32-bit) are needed and 8-bit lock register is needed.

Table 9-3. TZBMPCx describes the characteristics of the available TZBMPCx.

Table 9-3. TZBMPCx

MPC	Block	Number of blocks (y)	Kind of memory interface
TZBMPC3	Block based (block size = 256 bytes)	768 (0~23)	SRAM3 (192KB)
TZBMPC2	Block based (block size = 256 bytes)	512 (0~15)	SRAM2 (128KB)
TZBMPC1	Block based (block size = 256 bytes)	256 (0~7)	SRAM1 (64KB)
TZBMPC0	Block based (block size = 256 bytes)	256 (0~7)	SRAM0 (64KB)

If the interrupt be enabled by TZIAC, an interrupt signal is asserted whenever a security violation is detected. The interrupt is cleared by writing to the appropriate register.

9.3.5. TrustZone® illegal access controller (TZIAC)

TZIAC is secure peripherals, thus systematically generates an illegal access event when accessed by a non-secure access. It is used only when TrustZone® enabled (TZEN = 1). Correct settings of TZIAC allows the capture of the associated event and then generates the TZIACILA_IT interrupt to the NVIC. This applies for read, write and execute access. TZIAC can trace which event has triggered the NVIC TZIAC_S_IRQn. Register

`TZPCU_TZIAC_INTENx` can be used to enable an illegal access event, if an illegal access event happened, the flag bit in `TZPCU_TZIAC_STATx` register will be set, `TZPCU_TZIAC_STATCx` registers can use to clear an illegal access event flag bit.

9.3.6. SPC/GSSA debug

The DBGEN, SPIDEN, NIDEN and SPNIDEN signals determine the trace and debug state of secure and non-secure code. [Table 9-4. Trace and debug state](#) shows the relationship between four signals and the state of trace and debug.

Table 9-4. Trace and debug state

signals		trace and debug state
DBGEN	0	Secure and non-secure state debug is disabled
	1	Non-secure state debug is enabled
SPIDEN	0	Secure state debug is disabled
	1	Secure state debug is enabled
NIDEN	0	Secure and non-secure state trace is disabled
	1	Non-secure state trace is enabled
SPNIDEN	0	Secure state trace is disabled
	1	Secure state trace is enabled

The state of the signals is set according to the security protection (SPC) level and `TZPCU_TZSPC_DBG_CFG` register, security protection level has a high priority. The reset value of DBGEN, NIDEN, SPIDEN, and SPNIDEN is 1. If TZEN=0, DBGEN and NIDEN are automatically set to 1, and SPIDEN and SPNIDEN are automatically set to 0. If TZEN=1, then DBGEN, NIDEN, SPIDEN, and SPNIDEN are configured through the secure register. The DBGPAM bits in the `TZPCU_TZSPC_PAM_CFGx` is used to control the privilege access mode of `TZPCU_TZSPC_DBG_CFG` register, and this bit can be read or written only by secure privilege. In security protection 1 mode, debug cannot access flash, program QSPI, and read-protected sram. [Table 9-5. SPC/GSSA debug](#) shows the relationship between SPC/GSSA and debug.

Table 9-5. SPC/GSSA debug

	DBGEN	NIDEN	SPIDEN	SPNIDEN
No protection	Register configuration	Register configuration	Register configuration	Register configuration
Protection level 0.5	Register configuration	Register configuration	Force 0	Force 0
Protection level 1	Register configuration	Register configuration	Force 0	Force 0
GSSA	Force 0	Force 0	Force 0	Force 0

9.4. TZSPC Register definition

TZSPC secure access base address: 0x500A 0000

TZSPC non-secure access base address: 0x400A 0000

9.4.1. TZSPC control register (TZPCU_TZSPC_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

Secure write access only.

Read accesses are not limited.

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															

rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	LK	TZSPC items lock configuration bit This bit is set and cleared by software. 0: control register not locked 1: control register locked Note: This bit is unset by default and once set, it can not be reset until global TZSPC reset.

9.4.2. TZSPC secure access mode configuration register 0 (TZPCU_TZSPC_SAM_CFG0)

Address offset: 0x10

Reset value: 0x0000 0000

Secure write access only.

If a given bit in TZPCU_TZSPC_PAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written by non privilege secure code. If a given bit in TZPCU_TZSPC_PAM_CFGx register is set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written only by privilege secure code.

Read accesses are not limited.

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
SPI0SAM	TIMER0SAM	Reserved		USBFSSAM	Reserved										
rw	rw			rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1SAM	I2C0SAM	Reserved		USART2SAM	USART1SAM	Reserved	SPI1SAM	FWDGTSAM	WWDGTTSAM	Reserved	TIMER5SAM	TIMER4SAM	TIMER3SAM	TIMER2SAM	TIMER1SAM
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	SPI0SAM	SPI0 secure access mode configuration bit This bit is set and cleared by software. 0: Configure SPI0 secure access mode to non-secure 1: Configure SPI0 secure access mode to secure
30	TIMER0SAM	TIMER0 secure access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER0 secure access mode to non-secure 1: Configure TIMER0 secure access mode to secure
29:27	Reserved	Must be kept at reset value.
26	USBFSSAM	USBFS secure access mode configuration bit This bit is set and cleared by software. 0: Configure USBFS secure access mode to non-secure 1: Configure USBFS secure access mode to secure
25:16	Reserved	Must be kept at reset value.
15	I2C1SAM	I2C1 secure access mode configuration bit This bit is set and cleared by software. 0: Configure I2C1 secure access mode to non-secure 1: Configure I2C1 secure access mode to secure
14	I2C0SAM	I2C0 secure access mode configuration bit This bit is set and cleared by software. 0: Configure I2C0 secure access mode to non-secure 1: Configure I2C0 secure access mode to secure
13:12	Reserved	Must be kept at reset value
11	USART2SAM	USART2 secure access mode configuration bit This bit is set and cleared by software. 0: Configure USART2 secure access mode to non-secure

		1: Configure USART2 secure access mode to secure
10	USART1SAM	<p>USART1 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure USART1 secure access mode to non-secure</p> <p>1: Configure USART1 secure access mode to secure</p>
9	Reserved	Must be kept at reset value
8	SPI1SAM	<p>SPI1 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure SPI1 secure access mode to non-secure</p> <p>1: Configure SPI1 secure access mode to secure</p>
7	FWDGTSAM	<p>FWDGT secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure FWDGT secure access mode to non-secure</p> <p>1: Configure FWDGT secure access mode to secure</p>
6	WWDGTSAM	<p>WWDGT secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure WWDGT secure access mode to non-secure</p> <p>1: Configure WWDGT secure access mode to secure</p>
5	Reserved	Must be kept at reset value
4	TIMER5SAM	<p>TIMER5 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure TIMER5 secure access mode to non-secure</p> <p>1: Configure TIMER5 secure access mode to secure</p>
3	TIMER4SAM	<p>TIMER4 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure TIMER4 secure access mode to non-secure</p> <p>1: Configure TIMER4 secure access mode to secure</p>
2	TIMER3SAM	<p>TIMER3 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure TIMER3 secure access mode to non-secure</p> <p>1: Configure TIMER3 secure access mode to secure</p>
1	TIMER2SAM	<p>TIMER2 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure TIMER2 secure access mode to non-secure</p> <p>1: Configure TIMER2 secure access mode to secure</p>
0	TIMER1SAM	<p>TIMER1 secure access mode configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Configure TIMER1 secure access mode to non-secure</p>

1: Configure TIMER1 secure access mode to secure

9.4.3. TZSPC secure access mode configuration register 1 (TZPCU_TZSPC_SAM_CFG1)

Address offset: 0x014

Reset value: 0x0000 0000

Secure write access only.

If a given bit in TZPCU_TZSPC_PAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written by non privilege secure code. If a given bit in TZPCU_TZSPC_PAM_CFGx register is set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written only by privilege secure code.

Read accesses are not limited.

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															SDIOSA M
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKCAUS AM	TRNGSA M	HAUSAM	CAUSAM	ADCSAM	ICACHES AM	TSISAM	CRCSAM	HPDFSA M	Reserved	TIMER16 SAM	TIMER15 SAM	Reserved	USART0 SAM	Reserved	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	SDIOSAM	SDIO secure access mode configuration bit This bit is set and cleared by software. 0: Configure SDIO secure access mode to non-secure 1: Configure SDIO secure access mode to secure
15	PKCAUSAM	PKCAU secure access mode configuration bit This bit is set and cleared by software. 0: Configure PKCAU secure access mode to non-secure 1: Configure PKCAU secure access mode to secure
14	TRNGSAM	TRNG secure access mode configuration bit This bit is set and cleared by software. 0: Configure TRNG secure access mode to non-secure 1: Configure TRNG secure access mode to secure
13	HAUSAM	HAU secure access mode configuration bit

		This bit is set and cleared by software.
		0: Configure HAU secure access mode to non-secure
		1: Configure HAU secure access mode to secure
12	CAUSAM	CAU secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure CAU secure access mode to non-secure
		1: Configure CAU secure access mode to secure
11	ADCSAM	ADC secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure ADC secure access mode to non-secure
		1: Configure ADC secure access mode to secure
10	ICACHESAM	ICACHE secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure ICACHE secure access mode to non-secure
		1: Configure ICACHE secure access mode to secure
9	TSISAM	TSI secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure TSI secure access mode to non-secure
		1: Configure TSI secure access mode to secure
8	CRCSAM	CRC secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure CRC secure access mode to non-secure
		1: Configure CRC secure access mode to secure
7	HPDFSAM	HPDF secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure HPDF secure access mode to non-secure
		1: Configure HPDF secure access mode to secure
6:5	Reserved	Must be kept at reset value
4	TIMER16SAM	TIMER16 secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure TIMER16 secure access mode to non-secure
		1: Configure TIMER16 secure access mode to secure
3	TIMER15SAM	TIMER15 secure access mode configuration bit
		This bit is set and cleared by software.
		0: Configure TIMER15 secure access mode to non-secure
		1: Configure TIMER15 secure access mode to secure
2	Reserved	Must be kept at reset value
1	USART0SAM	USART0 secure access mode configuration bit

This bit is set and cleared by software.

0: Configure USART0 secure access mode to non-secure

1: Configure USART0 secure access mode to secure

0	Reserved	Must be kept at reset value
---	----------	-----------------------------

9.4.4. TZSPC secure access mode configuration register 2

(TZPCU_TZSPC_SAM_CFG2)

Address offset: 0x018

Reset value: 0x0000 0000

Secure write access only.

If a given bit in TZPCU_TZSPC_PAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written by non privilege secure code. If a given bit in TZPCU_TZSPC_PAM_CFGx register is set, the relative bit in TZPCU_TZSPC_SAM_CFGx register can be written only by privilege secure code.

Read accesses are not limited.

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
WIFISAM	DCISAM	I2S1_AD DSAM	WIFI_RF SAM	QSPI_FL ASHREG SAM	SQPI_PS RAMREG SAM	Reserved	EFUSES AM	Reserved							
rw	rw	rw	rw	rw	rw			rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31	WIFISAM	Wi-Fi secure access mode configuration bit This bit is set and cleared by software. 0: Configure Wi-Fi secure access mode to non-secure 1: Configure Wi-Fi secure access mode to secure
30	DCISAM	DCI secure access mode configuration bit This bit is set and cleared by software. 0: Configure DCI secure access mode to non-secure 1: Configure DCI secure access mode to secure
29	I2S1_ADDSAM	I2S1_ADD secure access mode configuration bit This bit is set and cleared by software.

			0: Configure I2S1_ADD secure access mode to non-secure 1: Configure I2S1_ADD secure access mode to secure
28	WIFI_RFSAM	Wi-Fi RF secure access mode configuration bit This bit is set and cleared by software.	0: Configure WIFI_RF secure access mode to non-secure 1: Configure WIFI_RF secure access mode to secure
27	QSPI_FLASHREGSA M	QSPI flash register secure access mode configuration bit This bit is set and cleared by software.	0: Configure QSPI flash register secure access mode to non-secure 1: Configure QSPI flash register secure access mode to secure
26	SQPI_PSRAMREGS AM	SQPI PSRAM register secure access mode configuration bit This bit is set and cleared by software.	0: Configure SQPI PSRAM register secure access mode to non-secure 1: Configure SQPI PSRAM flash register secure access mode to secure
25:24	Reserved	Must be kept at reset value.	
23	EFUSESAM	EFUSE register secure access mode configuration bit This bit is set and cleared by software.	0: Configure EFUSE register secure access mode to non-secure 1: Configure EFUSE register secure access mode to secure
22:0	Reserved	Must be kept at reset value.	

9.4.5. TZSPC privilege access mode configuration register 0

(TZPCU_TZSPC_PAM_CFG0)

Address offset: 0x20

Reset value: 0x0000 0000

Privilege write access only.

If a given bit in TZPCU_TZSPC_SAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written by non-secure privilege code. If a given bit in TZPCU_TZSPC_SAM_CFGx register is set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written only by secure privilege code.

Read accesses are not limited.

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
SPIOPAM rw	TIMEROP AM rw	Reserved		USBFSP AM rw							Reserved				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1PAM	I2C0PAM	Reserved		USART2 PAM	USART1 PAM	Reserved	SPI1PAM	FWDGTP AM	WWDGTT PAM	Reserved	TIMER5P AM	TIMER4P AM	TIMER3P AM	TIMER2P AM	TIMER1P AM
RW	RW			RW	RW		RW	RW	RW		RW	RW	RW	RW	RW

Bits	Fields	Descriptions
31	SPI0PAM	SPI0 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure SPI0 privilege access mode to non-privilege 1: Configure SPI0 privilege access mode to privilege
30	TIMER0PAM	TIMER0 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER0 privilege access mode to non-privilege 1: Configure TIMER0 privilege access mode to privilege
29:27	Reserved	Must be kept at reset value.
26	USBFSPAM	USBFS privilege access mode configuration bit This bit is set and cleared by software. 0: Configure USBFS privilege access mode to non-privilege 1: Configure USBFS privilege access mode to privilege
25:16	Reserved	Must be kept at reset value.
15	I2C1PAM	I2C1 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure I2C1 privilege access mode to non-privilege 1: Configure I2C1 privilege access mode to privilege
14	I2C0PAM	I2C0 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure I2C0 privilege access mode to non-privilege 1: Configure I2C0 privilege access mode to privilege
13:12	Reserved	Must be kept at reset value
11	USART2PAM	USART2 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure USART2 privilege access mode to non-privilege 1: Configure USART2 privilege access mode to privilege
10	USART1PAM	USART1 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure USART1 privilege access mode to non-privilege 1: Configure USART1 privilege access mode to privilege
9	Reserved	Must be kept at reset value

8	SPI1PAM	SPI1 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure SPI1 privilege access mode to non-privilege 1: Configure SPI1 privilege access mode to privilege
7	FWDGTPAM	FWDGT privilege access mode configuration bit This bit is set and cleared by software. 0: Configure FWDGT privilege access mode to non-privilege 1: Configure FWDGT privilege access mode to privilege
6	WWDGTPAM	WWDGTR privilege access mode configuration bit This bit is set and cleared by software. 0: Configure WWDGTR privilege access mode to non-privilege 1: Configure WWDGTR privilege access mode to privilege
5	Reserved	Must be kept at reset value
4	TIMER5PAM	TIMER5 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER5 privilege access mode to non-privilege 1: Configure TIMER5 privilege access mode to privilege
3	TIMER4PAM	TIMER4 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER4 privilege access mode to non-privilege 1: Configure TIMER4 privilege access mode to privilege
2	TIMER3PAM	TIMER3 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER3 privilege access mode to non-privilege 1: Configure TIMER3 privilege access mode to privilege
1	TIMER2PAM	TIMER2 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER2 privilege access mode to non-privilege 1: Configure TIMER2 privilege access mode to privilege
0	TIMER1PAM	TIMER1 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER1 privilege access mode to non-privilege 1: Configure TIMER1 privilege access mode to privilege

9.4.6. **TZSPC privilege access mode configuration register 1** **(TZPCU_TZSPC_PAM_CFG1)**

Address offset: 0x024

Reset value: 0x0000 0000

Privilege write access only.

If a given bit in TZPCU_TZSPC_SAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written by non-secure privilege code. If a given bit in TZPCU_TZSPC_SAM_CFGx register is set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written only by secure privilege code.

Read accesses are not limited.

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															SDIOPA M
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKCAUP AM	TRNGPA M	HAUPAM	CAUPAM	ADCPAM	ICACHEP AM	TSIPAM	CRCPAM	HPDFPA M	Reserved	TIMER16 PAM	TIMER15 PAM	Reserved	USART0 PAM	Reserved	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	SDIOPA M	SDIO privilege access mode configuration bit This bit is set and cleared by software. 0: Configure SDIO privilege access mode to non-privilege 1: Configure SDIO privilege access mode to privilege
15	PKCAUPA M	PKCAU privilege access mode configuration bit This bit is set and cleared by software. 0: Configure PKCAU privilege access mode to non-privilege 1: Configure PKCAU privilege access mode to privilege
14	TRNGPA M	TRNG privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TRNG privilege access mode to non-privilege 1: Configure TRNG privilege access mode to privilege
13	HAUPAM	HAU privilege access mode configuration bit This bit is set and cleared by software. 0: Configure HAU privilege access mode to non-privilege 1: Configure HAU privilege access mode to privilege
12	CAUPAM	CAU privilege access mode configuration bit This bit is set and cleared by software. 0: Configure CAU privilege access mode to non-privilege 1: Configure CAU privilege access mode to privilege

11	ADCPAM	ADC privilege access mode configuration bit This bit is set and cleared by software. 0: Configure ADC privilege access mode to non-privilege 1: Configure ADC privilege access mode to privilege
10	ICACHEPAM	ICACHE register privilege access mode configuration bit This bit is set and cleared by software. 0: Configure ICACHE register privilege access mode to non-privilege 1: Configure ICACHE register privilege access mode to privilege
9	TSIPAM	TSI privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TSI privilege access mode to non-privilege 1: Configure TSI privilege access mode to privilege
8	CRCPAM	CRC privilege access mode configuration bit This bit is set and cleared by software. 0: Configure CRC privilege access mode to non-privilege 1: Configure CRC privilege access mode to privilege
7	HPDFPAM	HPDF privilege access mode configuration bit This bit is set and cleared by software. 0: Configure HPDF privilege access mode to non-privilege 1: Configure HPDF privilege access mode to privilege
6:5	Reserved	Must be kept at reset value
4	TIMER16PAM	TIMER16 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER16 privilege access mode to non-privilege 1: Configure TIMER16 privilege access mode to privilege
3	TIMER15PAM	TIMER15 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure TIMER15 privilege access mode to non-privilege 1: Configure TIMER15 privilege access mode to privilege
2	Reserved	Must be kept at reset value
1	USART0PAM	USART0 privilege access mode configuration bit This bit is set and cleared by software. 0: Configure USART0 privilege access mode to non-privilege 1: Configure USART0 privilege access mode to privilege
0	Reserved	Must be kept at reset value.

9.4.7. TZSPC privilege access mode configuration register 2
(TZPCU_TZSPC_PAM_CFG2)

Address offset: 0x028

Reset value: 0x0000 0000

Privilege write access only

If a given bit in TZPCU_TZSPC_SAM_CFGx register is not set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written by non-secure privilege code. If a given bit in TZPCU_TZSPC_SAM_CFGx register is set, the relative bit in TZPCU_TZSPC_PAM_CFGx register can be written only by secure privilege code.

Read accesses are not limited.

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
WIFIPAM	DCIPAM	I2S1_AD DPAM	WIFI_RF PAM	QSPI_FL ASHREG PAM	SQPI_PS RAMREG PAM	DBGPM	Reserved	EFUSEP AM							Reserved
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31	WIFIPAM	Wi-Fi privilege access mode configuration bit This bit is set and cleared by software. 0: Configure Wi-Fi privilege access mode to non-privileged 1: Configure Wi-Fi privilege access mode to privileged
30	DCIPAM	DCI privilege access mode configuration bit This bit is set and cleared by software. 0: Configure DCI privilege access mode to non-privileged 1: Configure DCI privilege access mode to privileged
29	I2S1_ADDPA M	I2S1_ADD privilege access mode configuration bit This bit is set and cleared by software. 0: Configure I2S1_ADD privilege access mode to non-privileged 1: Configure I2S1_ADD privilege access mode to privileged
28	WIFI_RFPAM	Wi-Fi RF privilege access mode configuration bit This bit is set and cleared by software. 0: Configure Wi-Fi RF privilege access mode to non-privileged

			1: Configure Wi-Fi RF privilege access mode to privilege
27	QSPI_FLASHREGPA	QSPI flash register privilege access mode configuration bit	
	M	This bit is set and cleared by software.	
	0:	Configure QSPI flash register privilege access mode to non-privilege	
	1:	Configure QSPI flash register privilege access mode to privilege	
26	SQPI_PSRAMREGP	SQPI PSRAM register privilege access mode configuration bit	
	AM	This bit is set and cleared by software.	
	0:	Configure SQPI PSRAM register privilege access mode to non-privilege	
	1:	Configure SQPI PSRAM register privilege access mode to privilege	
25	DBGPAM	DBG register privilege access mode configuration bit	
		This bit is set and cleared by software.	
	0:	Configure DBG register privilege access mode to non-privilege	
	1:	Configure DBG register privilege access mode to privilege	
24	Reserved	Must be kept at reset value.	
23	EFUSEPAM	EFUSE register privilege access mode configuration bit	
		This bit is set and cleared by software.	
	0:	Configure EFUSE register privilege access mode to non-privilege	
	1:	Configure EFUSE register privilege access mode to privilege	
22:0	Reserved	Must be kept at reset value.	

9.4.8. TZSPC external memory x non-secure mark register 0 (TZPCU_TZSPC_TZMMPCx_NS0)

Address offset: $0x030 + 0x010 * x$, ($x = 0$ to 1)

Reset value: 0x0000 0000

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x4000 0000.

Secure access only.

NOTE: When NSM0_SADD + NSM0_LEN is over the maximum size of the memory, NSM0_LEN will set a constrained maximum.

Every TZPCU_TZMMPCx_NSMy($y=0\sim3$) register can define a non-secure area of the memory, the whole non-secure area of the memory is the union of these areas.

If NSM0_LEN = 16384 and TZPCU_NS0_SADD = 0, the whole 128-Mbyte memory space is non-secure (independent of TZPCU_TZMMPCx_NS1, TZPCU_TZMMPCx_NS2 and TZPCU_TZMMPCx_NS3 value).

If NSM0_LEN = 0x001 and NSM0_SADD = 0x3FF, only one 8-Kbyte block is defined as non-secure (at address offset = 0x7FF E000, ending at 0x07FF 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	NSM0_LEN[14:0]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NSM0_SADD[13:0]														

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:16	NSM0_LEN[14:0]	Length of the non-secure area (multiple of 8 Kbytes) of TZBMP0 Note: If NSM0_LEN + NSM0_SADD is over 16384, the value will set to 0x4000 – NSM0_SADD.
15:14	Reserved	Must be kept at reset value
13:0	NSM0_SADD[13:0]	The non-secure area (multiple of 8 Kbytes) start address of TZBMP0. Note: After reset, if TZEN=1, the memory is marked to secure, if TZEN=0, the memory is marked to non-secure.

9.4.9. TZSPC external memory x non-secure mark register 1

(TZPCU_TZSPC_TZMMPCx_NSMy)

Address offset: 0x034 + 0x010 * x, (x = 0 to 1)

Reset value: 0x0000 0000

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x4000 0000.

Secure access only.

NOTE: When NSM1_SADD + NSM1_LEN is over the maximum size of the memory, NSM1_LEN will set a constrained maximum.

Every TZPCU_TZMMPCx_NSMy(y=0~3) register can define a non-secure area of the memory, the whole non-secure area of the memory is the union of these areas.

If NSM0_LEN = 16384 and TZPCU_NS0_SADD = 0, the whole 128-Mbyte memory space is non-secure (independent of TZPCU_TZMMPCx_NS1, TZPCU_TZMMPCx_NS2 and TZPCU_TZMMPCx_NS3 value).

If NSM0_LEN = 0x001 and NSM0_SADD = 0x3FF, only one 8-Kbyte block is defined as non-secure (at address offset = 0x7FF E000, ending at 0x07FF 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	NSM1_LEN[14:0]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NSM1_SADD[13:0]														

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:16	NSM1_LEN[14:0]	Length of the non-secure area (multiple of 8 Kbytes) of TZBMP1 Note: If NSM1_LEN + NSM1_SADD is over 16384, the value will set to 0x4000 – NSM1_SADD.
15:14	Reserved	Must be kept at reset value
13:0	NSM1_SADD[13:0]	The non-secure area (multiple of 8 Kbytes) start address of TZBMP1. Note: After reset, if TZEN=1, the memory is marked to secure, if TZEN=0, the memory is marked to non-secure.

9.4.10. TZSPC external memory x non-secure mark register 2

(TZPCU_TZSPC_TZMMPCx_NSMy)

Address offset: 0x038 + 0x010 * x, (x = 0)

Reset value: 0x0000 0000

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x4000 0000.

Secure access only.

NOTE: When NSM2_SADD + NSM2_LEN is over the maximum size of the memory, NSM2_LEN will set a constrained maximum.

Every TZPCU_TZMMPCx_NSMy(y=0~3) register can define a non-secure area of the memory, the whole non-secure area of the memory is the union of these areas.

If NSM0_LEN = 16384 and TZPCU_NS0_SADD = 0, the whole 128-Mbyte memory space is non-secure (independent of TZPCU_TZMMPCx_NS1, TZPCU_TZMMPCx_NS2 and TZPCU_TZMMPCx_NS3 value).

If NSM0_LEN = 0x001 and NSM0_SADD = 0x3FF, only one 8-Kbyte block is defined as non-secure (at address offset = 0x7FF E000, ending at 0x07FF 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	NSM2_LEN[14:0]														

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:16	NSM2_LEN[14:0]	Length of the non-secure area (multiple of 8 Kbytes) of TZBMPC2 Note: If NSM0_LEN + NSM2_SADD is over 16384, the value will set to 0x4000 – NSM2_SADD.
15:14	Reserved	Must be kept at reset value
13:0	NSM2_SADD[13:0]	The non-secure area (multiple of 8 Kbytes) start address of TZBMPC2. Note: After reset, if TZEN=1, the memory is marked to secure, if TZEN=0, the memory is marked to non-secure.

9.4.11. TZSPC external memory x non-secure mark register 3

(TZPCU TZSPC TZMMPCx NSM3)

Address offset: 0x03C + 0x010 * x, (x = 0)

Reset value: 0x0000 0000

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x4000 0000.

Secure access only.

NOTE: When NSM3_SADD + NSM3_LEN is over the maximum size of the memory, NSM3_LEN will set a constrained maximum.

Every TZPCU_TZMMPCx_NSMy(y=0~3) register can define a non-secure area of the memory, the whole non-secure area of the memory is the union of these areas.

If NSM0_LEN = 16384 and TZPCU_NSM0_SADD = 0, the whole 128-Mbyte memory space is non-secure (independent of TZPCU_TZMMPCx_NS1, TZPCU_TZMMPCx_NS2 and TZPCU_TZMMPCx_NS3 value).

If NSM0_LEN = 0x001 and NSM0_SADD = 0x3FF, only one 8-Kbyte block is defined as non-secure (at address offset = 0x7FF E000, ending at 0x07FF 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	NSM3_LEN[14:0]														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	NSM3_SADD[13:0]
----------	-----------------

rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:16	NSM3_LEN[14:0]	Length of the non-secure area (multiple of 8 Kbytes) of TZBMPC3 Note: If NSM3_LEN + NSM3_SADD is over 16384, the value will set to 0x4000 – NSM3_SADD.
15:13	Reserved	Must be kept at reset value
13:0	NSM3_SADD[13:0]	The non-secure area (multiple of 8 Kbytes) start address of TZBMPC3. Note: After reset, if TZEN=1, the memory is marked to secure, if TZEN=0, the memory is marked to non-secure.

9.4.12. TZSPC debug configuration register (TZPCU_TZSPC_DBG_CFG)

Address offset: 0x200

Reset value: 0x0000 000F

Secure write access only.

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x0000 0003.

If DBGPAM bit in TZPCU_TZSPC_PAM_CFG is not set, this register can be written by non privilege secure code. If DBGPAM bit in TZPCU_TZSPC_PAM_CFG is set, this register can be written only by privilege secure code.

Read accesses are not limited.

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												SPNIDEN	SPIDEN	NIDEN	IDEN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	SPNIDEN	Secure non-invasive debug enable bit This bit is set and cleared by software. 0: Disabled secure non-invasive debug 1: Enabled secure non-invasive debug

2	SPIDEN	Secure invasive debug enable bit This bit is set and cleared by software. 0: Disabled secure invasive debug 1: Enabled secure invasive debug
1	NIDEN	Non-invasive debug enable bit This bit is set and cleared by software. 0: Disabled non-invasive debug 1: Enabled non-invasive debug
0	IDEN	Invasive debug enable bit(DBGEN bit) This bit is set and cleared by software. 0: Disabled invasive debug 1: Enabled invasive debug

9.5. TZBMP0 registers definition

TZBMP0 secure access base address: 0x500A 0800

TZBMP0 non-secure access base address: 0x400A 0800

9.5.1. TZBMP0 control register (TZPCU_TZBMP0_CTL)

Address offset: 0x000

Reset value: 0x0000 0000

Secure access only.

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
SRWACF	SECSTAT														
G	CFG														
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															LK
															rw

Bits	Fields	Descriptions
31	SRWACFG	Secure read/w rite access non-secure SRAM configuration bit. This bit is set and cleared by software. 0: Configure secure read/w rite access non-secure SRAM is illegal 1: Configure secure read/w rite access non-secure SRAM is legal
30	SECSTATCFG	Security state configuration bit.

This bit is set and cleared by software.

0: Configure TZBMPc source clock to non-secure if there do not exists secure area in TZBMPc, if exists secure area, then TZBMPc source clock is secure

1: Configure TZBMPc source clock still to secure if there do not exists secure area in TZBMPc

29:1	Reserved	Must be kept at reset value
0	LK	<p>The control register of the TZBMPc sub-block lock configuration bit</p> <p>This bit is set and cleared by software.</p> <p>0: Not lock control registe of the TZBMPc sub-block</p> <p>1: Lock control registe of the TZBMPc sub-block</p> <p>Note: This bit is unset by default and once set, it can not be reset until next reset.</p>

9.5.2. TZBMPc0 vector register y (TZPCU_TZBMPc0_VECy)

Address offset: 0x100 + 0x04 * y, (y = 0 to 7)

Reset value: 0xFFFF FFFF

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x0000 0000.

Secure access only.

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
B(31+32*y)	B(30+32*y)	B(29+32*y)	B(28+32*y)	B(27+32*y)	B(26+32*y)	B(25+32*y)	B(24+32*y)	B(23+32*y)	B(22+32*y)	B(21+32*y)	B(20+32*y)	B(19+32*y)	B(18+32*y)	B(17+32*y)	B(16+32*y)
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B(15+32*y)	B(14+32*y)	B(13+32*y)	B(12+32*y)	B(11+32*y)	B(10+32*y)	B(9+32*y)	B(8+32*y)	B(7+32*y)	B(6+32*y)	B(5+32*y)	B(4+32*y)	B(3+32*y)	B(2+32*y)	B(1+32*y)	B(0+32*y)
rw															

Bits	Fields	Descriptions
31:0	B (x + 32 * y)	<p>Secure access mode for the union-block y configuration bits x, x=0..31.</p> <p>These bits are set and cleared by software</p> <p>0: block of union-block y is non-secure.</p> <p>1: block of union-block y is secure.</p>

9.5.3. TZBMPc0 lock register 0 (TZPCU_TZBMPc0_LOCK0)

Address offset: 0x010

Reset value: 0x0000 0000

Secure access only.

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

Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LKUB7	LKUB6	LKUB5	LKUB4	LKUB3	LKUB2	LKUB1	LKUB0

rwo rwo

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	LKUB[7:0]	The union-blocks 0 to 7 secure access mode lock configuration bits. These bits are set and cleared by software 0x0000 0000: security configuration unlocked for all union-blocks 0x0000 0001: security configuration locked only for union-blocks 0 0x0000 000F: security configuration locked for all union-blocks of SRAM0

9.6. TZBMP1 registers definition

TZBMP1 secure access base address: 0x500A 0C00

TZBMP1 non-secure access base address: 0x400A 0C00

9.6.1. TZBMP1 control register (TZPCU_TZBMP1_CTL)

Address offset: 0x000

Reset value: 0x0000 0000

Secure access only.

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
SRWACF	SECSTAT														
G	CFG														

rw rw

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

Reserved

Reserved	LK
----------	----

rw

Bits	Fields	Descriptions
31	SRWACFG	Secure read/w rite access non-secure SRAM configuration bit. This bit is set and cleared by softw are. 0: Configure secure read/w rite access non-secure SRAM is illegal 1: Configure secure read/w rite access non-secure SRAM is legal
30	SECSTATCFG	Security state configuration bit. This bit is set and cleared by softw are. 0: Configure TZBMPC source clock to non-secure if there do not exists secure area in TZBMPC, if exits secure area, then TZBMPC source clock is secure 1: Configure TZBMPC source clock still to secure if there do not exists secure area in TZBMPC
29:1	Reserved	Must be kept at reset value
0	LK:	The control register of the TZBMPC sub-block lock configuration bit This bit is set and cleared by softw are. 0: Not lock control registe of the TZBMPC sub-block 1: Lock control registe of the TZBMPC sub-block Note: This bit is unset by default and once set, it can not be reset until next reset.

9.6.2. TZBMPC1 vector register y (TZPCU_TZBMPC1_VECy)

Address offset: 0x100 + 0x04 * y, (y = 0 to 7)

Reset value: 0xFFFF FFFF

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x0000 0000.

Secure access only.

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
B(31+32*y)	B(30+32*y)	B(29+32*y)	B(28+32*y)	B(27+32*y)	B(26+32*y)	B(25+32*y)	B(24+32*y)	B(23+32*y)	B(22+32*y)	B(21+32*y)	B(20+32*y)	B(19+32*y)	B(18+32*y)	B(17+32*y)	B(16+32*y)
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B(15+32*y)	B(14+32*y)	B(13+32*y)	B(12+32*y)	B(11+32*y)	B(10+32*y)	B(9+32*y)	B(8+32*y)	B(7+32*y)	B(6+32*y)	B(5+32*y)	B(4+32*y)	B(3+32*y)	B(2+32*y)	B(1+32*y)	B(0+32*y)
rw															

Bits	Fields	Descriptions
-------------	---------------	---------------------

31:0	B (x + 32 * y)	Secure access mode for the union-block y configuration bits x, x=0..31. These bits are set and cleared by software 0: block of union-block y is non-secure. 1: block of union-block y is secure.
------	----------------	---

9.6.3. TZBMPc1 lock register 0 (TZPCU_TZBMPc1_LOCK0)

Address offset: 0x010

Reset value: 0x0000 0000

Secure access only.

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

Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LKUB7	LKUB6	LKUB5	LKUB4	LKUB3	LKUB2	LKUB1	LKUB0
rwo								rwo							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	LKUB[7:0]	The union-blocks 0 to 7 secure access mode lock configuration bits. These bits are set and cleared by software 0x0000 0000: security configuration unlocked for all union-blocks 0x0000 0001: security configuration locked only for union-blocks 0 0x0000 000F: security configuration locked for all union-blocks of SRAM1

9.7. TZBMPc2 registers definition

TZBMPc2 secure access base address: 0x500B 0000

TZBMPc2 non-secure access base address: 0x400B 0000

9.7.1. TZBMPc2 control register (TZPCU_TZBMPc2_CTL)

Address offset: 0x000

Reset value: 0x0000 0000

Secure access only.

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
SRWACF	SECSTAT														
G	CFG														Reserved
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															Reserved
															LK
															rw

Bits	Fields	Descriptions
31	SRWACFG	Secure read/w rite access non-secure SRAM configuration bit. This bit is set and cleared by software. 0: Configure secure read/w rite access non-secure SRAM is illegal 1: Configure secure read/w rite access non-secure SRAM is legal
30	SECSTATCFG	Security state configuration bit. This bit is set and cleared by software. 0: Configure TZBMPC source clock to non-secure if there do not exists secure area in TZBMPC, if exits secure area, then TZBMPC source clock is secure 1: Configure TZBMPC source clock still to secure if there do not exists secure area in TZBMPC
29:1	Reserved	Must be kept at reset value
0	LK	The control register of the TZBMPC sub-block lock configuration bit This bit is set and cleared by software. 0: Not lock control registe of the TZBMPC sub-block 1: Lock control registe of the TZBMPC sub-block Note: This bit is unset by default and once set, it can not be reset until next reset.

9.7.2. TZBMPC2 vector register y (TZPCU_TZBMPC2_VECy)

Address offset: 0x100 + 0x04 * y, (y = 0 to 15)

Reset value: 0xFFFF FFFF

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x0000 0000.

Secure access only.

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
B(31+32* y)	B(30+32* y)	B(29+32* y)	B(28+32* y)	B(27+32* y)	B(26+32* y)	B(25+32* y)	B(24+32* y)	B(23+32* y)	B(22+32* y)	B(21+32* y)	B(20+32* y)	B(19+32* y)	B(18+32* y)	B(17+32* y)	B(16+32* y)

rw	rw	rw	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			
B(15+32*y)	B(14+32*y)	B(13+32*y)	B(12+32*y)	B(11+32*y)	B(10+32*y)	B(9+32*y)	B(8+32*y)	B(7+32*y)	B(6+32*y)	B(5+32*y)	B(4+32*y)	B(3+32*y)	B(2+32*y)	B(1+32*y)	B(0+32*y)			

Bits	Fields	Descriptions
31:0	B (x + 32 * y)	<p>Secure access mode for the union-block y configuration bits x, x=0..31.</p> <p>These bits are set and cleared by software</p> <p>0: block of union-block y is non-secure.</p> <p>1: block of union-block y is secure.</p>

9.7.3. TZBMPC2 lock register 0 (TZPCU_TZBMPC2_LOCK0)

Address offset: 0x010

Reset value: 0x0000 0000

Secure access only.

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

Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	LKUB[15:0]	<p>The union-blocks 0 to 15 secure access mode lock configuration bits.</p> <p>These bits are set and cleared by software</p> <p>0x0000 0000: security configuration unlocked for all union-blocks</p> <p>0x0000 0001: security configuration locked only for union-blocks 0</p> <p>....</p> <p>0x0000 00FF: security configuration locked for all union-blocks of SRAM2</p>

9.8. TZBMPc3 registers definition

TZBMP3 secure access base address: 0x500B 0400

TZBMPC3 non-secure access base address: 0x400B 0400

9.8.1. TZBMPC3 control register (TZPCU_TZBMPC3_CTL)

Address offset: 0x000

Reset value: 0x0000 0000

Secure access only.

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
SRWACF	SECSTAT														
G	CFG														Reserved
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															Reserved
															rw

Bits	Fields	Descriptions
31	SRWACFG	Secure read/w rite access non-secure SRAM configuration bit. This bit is set and cleared by softw are. 0: Configure secure read/w rite access non-secure SRAM is illegal 1: Configure secure read/w rite access non-secure SRAM is legal
30	SECSTATCFG	Security state configuration bit. This bit is set and cleared by softw are. 0: Configure TZBMPC source clock to non-secure if there do not exists secure area in TZBMPC, if exits secure area, then TZBMPC source clock is secure 1: Configure TZBMPC source clock still to secure if there do not exists secure area in TZBMPC
29:1	Reserved	Must be kept at reset value
0	LK	The control register of the TZBMPC sub-block lock configuration bit This bit is set and cleared by softw are. 0: Not lock control registe of the TZBMPC sub-block 1: Lock control registe of the TZBMPC sub-block Note: This bit is unset by default and once set, it can not be reset until next reset.

9.8.2. TZBMPC3 vector register y (TZPCU_TZBMPC3_VECy)

Address offset: 0x100 + 0x04 * y, (y = 0 to 23)

Reset value: 0xFFFF FFFF

If TZEN = 1, the given reset value is valid.

If TZEN = 0, the reset value is 0x0000 0000.

Secure access only.

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
B(31+32*y)	B(30+32*y)	B(29+32*y)	B(28+32*y)	B(27+32*y)	B(26+32*y)	B(25+32*y)	B(24+32*y)	B(23+32*y)	B(22+32*y)	B(21+32*y)	B(20+32*y)	B(19+32*y)	B(18+32*y)	B(17+32*y)	B(16+32*y)
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B(15+32*y)	B(14+32*y)	B(13+32*y)	B(12+32*y)	B(11+32*y)	B(10+32*y)	B(9+32*y)	B(8+32*y)	B(7+32*y)	B(6+32*y)	B(5+32*y)	B(4+32*y)	B(3+32*y)	B(2+32*y)	B(1+32*y)	B(0+32*y)
rw															

Bits	Fields	Descriptions
31:0	B ([31:0]+ 32 * y)	Secure access mode for the union-block y configuration bits. These bits are set and cleared by software 0x0000 0000: all blocks of union-block y are non-secure. 0x0000 0001: only blocks 0 of union-block y are secure. 0xFFFF FFFF: all union-blocks are secure.

9.8.3. TZBMP3 lock register 0 (TZPCU_TZBMP3_LOCK0)

Address offset: 0x010

Reset value: 0x0000 0000

Secure access only.

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

Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								LKUB23	LKUB22	LKUB21	LKUB20	LKUB19	LKUB18	LKUB17	LKUB16
								rwo							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LKUB15	LKUB14	LKUB13	LKUB12	LKUB11	LKUB10	LKUB9	LKUB8	LKUB7	LKUB6	LKUB5	LKUB4	LKUB3	LKUB2	LKUB1	LKUB0
rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	LKUB[23:0]	The union-blocks 0 to 23 secure access mode lock configuration bits. These bits are set and cleared by software 0x0000 0000: security configuration unlocked for all union-blocks

0x0000 0001: security configuration locked only for union-blocks 0

....

0x0000 0FFF: security configuration locked for all union-blocks of SRAM3

9.9. TZIAC Register definition

TZIAC secure access base address: 0x500A 0400

TZIAC non-secure access base address: 0x400A 0400

9.9.1. TZIAC interrupt enable register 0 (TZPCU_TZIAC_INTEN0)

Address offset: 0x000

Reset value: 0x0000 0000

Secure access only.

This register is used to enable/disable illegal access event for each source.

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
SPI0IE	TIMER0IE		Reserved		USBFSIE										Reserved
rw	rw				rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1IE	I2C0IE		Reserved	USART2I	USART1I	Reserved	SPI1IE	FWDGTI	WWDGTI	Reserved	TIMER5I	TIMER4I	TIMER3I	TIMER2I	TIMER1I
rw	rw			rw	rw		rw	rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	SPI0IE	SPI0 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable SPI0 illegal access interrupt 1: Enable SPI0 illegal access interrupt
30	TIMER0IE	TIMER0 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER0 illegal access interrupt 1: Enable TIMER0 illegal access interrupt
29:27	Reserved	Must be kept at reset value.
26	USBFSIE	USBFS illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable USBFS illegal access interrupt

		1: Enable USBFS illegal access interrupt
25:16	Reserved	Must be kept at reset value.
15	I2C1IE	<p>I2C1 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable I2C1 illegal access interrupt</p> <p>1: Enable I2C1 illegal access interrupt</p>
14	I2C0IE	<p>I2C0 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable I2C0 illegal access interrupt</p> <p>1: Enable I2C0 illegal access interrupt</p>
13:12	Reserved	Must be kept at reset value
11	USART2IE	<p>USART2 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable USART2 illegal access interrupt</p> <p>1: Enable USART2 illegal access interrupt</p>
10	USART1IE	<p>USART1 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable USART1 illegal access interrupt</p> <p>1: Enable USART1 illegal access interrupt</p>
9	Reserved	Must be kept at reset value
8	SPI1IE	<p>SPI1 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable SPI1 illegal access interrupt</p> <p>1: Enable SPI1 illegal access interrupt</p>
7	FWDGTIE	<p>FWDGT illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable FWDGT illegal access interrupt</p> <p>1: Enable FWDGT illegal access interrupt</p>
6	WWDGTIE	<p>WWDGT illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable WWDGT illegal access interrupt</p> <p>1: Enable WWDGT illegal access interrupt</p>
5	Reserved	Must be kept at reset value
4	TIMER5IE	<p>TIMER5 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable TIMER5 illegal access interrupt</p> <p>1: Enable TIMER5 illegal access interrupt</p>

3	TIMER4IE	TIMER4 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER4 illegal access interrupt 1: Enable TIMER4 illegal access interrupt
2	TIMER3IE	TIMER3 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER3 illegal access interrupt 1: Enable TIMER3 illegal access interrupt
1	TIMER2IE	TIMER2 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER2 illegal access interrupt 1: Enable TIMER2 illegal access interrupt
0	TIMER1IE	TIMER1 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER1 illegal access interrupt 1: Enable TIMER1 illegal access interrupt

9.9.2. TZIAC interrupt enable register 1 (TZPCU_TZIAC_INTEN1)

Address offset: 0x004

Reset value: 0x0000 0000

Secure access only.

This register is used to enable/disable illegal access event for each source.

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	EXTIIE	FMCIE	FLASHIE	RCUIE	Reserved	DMA1IE	DMA0IE	SYSFCGI E	PMUIE	RTCIE	Reserved	SDIOIE			
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
PKCAUIE	TRNGIE	HAUIE	CAUIE	ADCIE	ICACHE1 E	TSIIIE	CRCIE	HPDFIE	Reserved	TIMER16I E	TIMER15I E	Reserved	USART0I E	Reserved	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	EXTIIE	EXTI illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable EXTI illegal access interrupt

		1: Enable EXTI illegal access interrupt
27	FMCIE	<p>FMC illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable FMC illegal access interrupt</p> <p>1: Enable FMC illegal access interrupt</p>
26	FLASHIE	<p>FLASH illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable FLASH illegal access interrupt</p> <p>1: Enable FLASH illegal access interrupt</p>
25	RCUIE	<p>RCU illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable RCU illegal access interrupt</p> <p>1: Enable RCU illegal access interrupt</p>
24	Reserved	Must be kept at reset value.
23	DMA1IE	<p>DMA1 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable DMA1 illegal access interrupt</p> <p>1: Enable DMA1 illegal access interrupt</p>
22	DMA0IE	<p>DMA0 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable DMA0 illegal access interrupt</p> <p>1: Enable DMA0 illegal access interrupt</p>
21	SYSCFGIE	<p>SYSCFG illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable SYSCFG illegal access interrupt</p> <p>1: Enable SYSCFG illegal access interrupt</p>
20	PMUIE	<p>PMU illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable PMU illegal access interrupt</p> <p>1: Enable PMU illegal access interrupt</p>
19	RTCIE	<p>RTC illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable RTC illegal access interrupt</p> <p>1: Enable RTC illegal access interrupt</p>
18:17	Reserved	Must be kept at reset value
16	SDIOIE	<p>SDIO illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p> <p>0: Disable SDIO illegal access interrupt</p>

		1: Enable SDIO illegal access interrupt
15	PKCAUIE	PKCAU illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable PKCAU illegal access interrupt 1: Enable PKCAU illegal access interrupt
14	TRNGIE	TRNG illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TRNG illegal access interrupt 1: Enable TRNG illegal access interrupt
13	HAUIE	HAU illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable HAU illegal access interrupt 1: Enable HAU illegal access interrupt
12	CAUIE	CAU illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable CAU illegal access interrupt 1: Enable CAU illegal access interrupt
11	ADCIE	ADC illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable ADC illegal access interrupt 1: Enable ADC illegal access interrupt
10	ICACHEIE	ICACHE illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable ICACHE illegal access interrupt 1: Enable ICACHE illegal access interrupt
9	TSIIE	TSI illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TSI illegal access interrupt 1: Enable TSI illegal access interrupt
8	CRCIE	CRC illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable CRC illegal access interrupt 1: Enable CRC illegal access interrupt
7	HPDFIE	HPDF illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable HPDF illegal access interrupt 1: Enable HPDF illegal access interrupt
6:5	Reserved	Must be kept at reset value

4	TIMER16IE	TIMER16 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER16 illegal access interrupt 1: Enable TIMER16 illegal access interrupt
3	TIMER15IE	TIMER15 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TIMER15 illegal access interrupt 1: Enable TIMER15 illegal access interrupt
2	Reserved	Must be kept at reset value
1	USART0IE	USART0 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable USART0 illegal access interrupt 1: Enable USART0 illegal access interrupt
0	Reserved	Must be kept at reset value

9.9.3. TZIAC interrupt enable register 2 (TZPCU_TZIAC_INTEN2)

Address offset: 0x008

Reset value: 0x0000 0000

Secure access only.

This register is used to enable/disable illegal access event for each source.

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
WIFIIE	DCIIIE	I2S1_AD DIE	WIFI_RFI E	QSPI_FL ASHREGI E	SQPI_PS RAMREG IE	QSPI_FL ASHIE	SQPI_PS RAMIE	EFUSEIE							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
Reserved				TZBMPC 3_REGIE	SRAM3IE	TZBMPC 2_REGIE	SRAM2IE	TZBMPC 1_REGIE	SRAM1IE	TZBMPC 0_REGIE	SRAM0IE	Reserved	TZIACIE	TZSPCIE	
				rw	rw	rw	rw	rw	rw	r	rw		rw	rw	

Bits	Fields	Descriptions
31	WIFIIE	Wi-Fi illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable Wi-Fi illegal access interrupt 1: Enable Wi-Fi illegal access interrupt
30	DCIIIE	DCI illegal access interrupt enable bit

		This bit is set and cleared by software. 0: Disable DCI illegal access interrupt 1: Enable DCI illegal access interrupt
29	I2S1_ADDIE	I2S1_ADD illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable I2S1_ADD illegal access interrupt 1: Enable I2S1_ADD illegal access interrupt
28	WIFI_RFIE	Wi-Fi RF illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable WIFI_RF illegal access interrupt 1: Enable WIFI_RF illegal access interrupt
27	QSPI_FLASHREGIE	QSPI FLASHREG illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable QSPI_FLASHREG illegal access interrupt 1: Enable QSPI_FLASHREG illegal access interrupt
26	SQPI_PSRAMREGIE	SQPI PSRAMREG illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable SQPI_PSRAMREG illegal access interrupt 1: Enable SQPI_PSRAMREG illegal access interrupt
25	QSPI_FLASHIE	QSPI FLASH illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable QSPI FLASH illegal access interrupt 1: Enable QSPI FLASH illegal access interrupt
24	SQPI_PSRAmie	SQPI PSRAM illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable SQPI PSRAM illegal access interrupt 1: Enable SQPI PSRAM illegal access interrupt
23	EFUSEIE	EFUSE illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable EFUSE illegal access interrupt 1: Enable EFUSE illegal access interrupt
22:12	Reserved	Must be kept at reset value.
11	TZBMP3_REGIE	TZBMP3 REG illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable TZBMP3 REG illegal access interrupt 1: Enable TZBMP3 REG illegal access interrupt
10	SRAM3IE	SRAM3 illegal access interrupt enable bit This bit is set and cleared by software. 0: Disable SRAM3 illegal access interrupt

		1: Eisable SRAM3 illegal access interrupt
9	TZBMP2_REGIE	<p>TZBMP2 REG illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable TZBMP2 REG illegal access interrupt
		1: Eisable TZBMP2 REG illegal access interrupt
8	SRAM2IE	<p>SRAM2 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable SRAM2 illegal access interrupt
		1: Eisable SRAM2 illegal access interrupt
7	TZBMP1_REGIE	<p>TZBMP1 REG illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable TZBMP1 REG illegal access interrupt
		1: Eisable TZBMP1 REG illegal access interrupt
6	SRAM1IE	<p>SRAM1 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable SRAM1 illegal access interrupt
		1: Eisable SRAM1 illegal access interrupt
5	TZBMP0_REGIE	<p>TZBMP0 REG illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable TZBMP0 REG illegal access interrupt
		1: Eisable TZBMP0 REG illegal access interrupt
4	SRAM0IE	<p>SRAM0 illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable SRAM0 illegal access interrupt
		1: Eisable SRAM0 illegal access interrupt
3:2	Reserved	Must be kept at reset value.
1	TZIACIE	<p>TZIAC illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable TZIAC illegal access interrupt
		1: Eisable TZIAC illegal access interrupt
0	TZSPCIE	<p>TZSPC illegal access interrupt enable bit</p> <p>This bit is set and cleared by software.</p>
		0: Disable TZSPC illegal access interrupt
		1: Eisable TZSPC illegal access interrupt

9.9.4. TZIAC status register 0 (TZPCU_TZIAC_STAT0)

Address offset: 0x010

Reset value: 0x0000 0000

Secure access only.

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
SPI0IAF	TIMER0I AF	Reserved		USBFSIA F	Reserved										
r	r			r											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1IAF	I2C0IAF	Reserved		USART2I AF	USART1I AF	Reserved	SPI1IAF	FWDGTI AF	WWDGTI AF	Reserved	TIMER5I AF	TIMER4I AF	TIMER3I AF	TIMER2I AF	TIMER1I AF
r	r			r	r		r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31	SPI0IAF	SPI0 illegal access event flag bit 0: no SPI0 illegal access event 1: SPI0 illegal access event pending
30	TIMER0IAF	TIMER0 illegal access event flag bit 0: no TIMER0 illegal access event 1: TIMER0 illegal access event pending
29:27	Reserved	Must be kept at reset value.
26	USBFSIAF	USBFS illegal access event flag bit 0: no USBFS illegal access event 1: USBFS illegal access event pending
25:16	Reserved	Must be kept at reset value.
15	I2C1IAF	I2C1 illegal access flag bit This bit is set and cleared by software. 0: Disable I2C1 illegal access interrupt 1: Enable I2C1 illegal access interrupt
14	I2C0IAF	I2C0 illegal access flag bit This bit is set and cleared by software. 0: Disable I2C0 illegal access interrupt 1: Enable I2C0 illegal access interrupt
13:12	Reserved	Must be kept at reset value
11	USART2IAF	USART2 illegal access event flag bit 0: no USART2 illegal access event 1: USART2 illegal access event pending
10	USART1IAF	USART1 illegal access event flag bit 0: no USART1 illegal access event

		1: USART1 illegal access event pending
9	Reserved	Must be kept at reset value
8	SPI1IAF	SPI1 illegal access event flag bit 0: no SPI1 illegal access event 1: SPI1 illegal access event pending
7	FWDGTIA F	FWDGT illegal access event flag bit 0: no FWDGT illegal access event 1: FWDGT illegal access event pending
6	WWDGTIAF	WWDGT illegal access event flag bit 0: no WWDGT illegal access event 1: WWDGT illegal access event pending
5	Reserved	Must be kept at reset value
4	TIMER5IA F	TIMER5 illegal access event flag bit 0: no TIMER5 illegal access event 1: TIMER5 illegal access event pending
3	TIMER4IA F	TIMER4 illegal access event flag bit 0: no TIMER4 illegal access event 1: TIMER4 illegal access event pending
2	TIMER3IA F	TIMER3 illegal access event flag bit 0: no TIMER3 illegal access event 1: TIMER3 illegal access event pending
1	TIMER2IA F	TIMER2 illegal access event flag bit 0: no TIMER2 illegal access event 1: TIMER2 illegal access event pending
0	TIMER1IA F	TIMER1 illegal access event flag bit 0: no TIMER1 illegal access event 1: TIMER1 illegal access event pending

9.9.5. TZIAC status register 1 (TZPCU_TZIAC_STAT1)

Address offset: 0x014

Reset value: 0x0000 0000

Secure access only.

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	EXTIIAF	FMCIAF	FLASHIA F	RCUIAF	Reserved	DMA1IAF	DMA0IAF	SYS CFGI AF	PMUIAF	RTClAF	Reserved	SDIOIAF			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKCAUIAF	TRNGIAF	HAUIAF	CAUIAF	ADCIAF	ICACHEIAF	TSIIAF	CRCIAF	HPDFIAF	Reserved	TIMER16IAF	TIMER15IAF	Reserved	USARTTOIAF	Reserved	

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	EXTIIAF	EXTI illegal access event flag bit 0: no EXTI illegal access event 1: EXTI illegal access event pending
27	FMCIAF	FMC illegal access event flag bit 0: no FMC illegal access event 1: FMC illegal access event pending
26	FLASHIAF	FLASH illegal access event flag bit 0: no FLASH illegal access event 1: FLASH illegal access event pending
25	RCUIAF	RCU illegal access event flag bit 0: no RCU illegal access event 1: RCU illegal access event pending
24	Reserved	Must be kept at reset value
23	DMA1IAF	DMA1 illegal access event flag bit 0: no DMA1 illegal access event 1: DMA1 illegal access event pending
22	DMA0IAF	DMA0 illegal access event flag bit 0: no DMA0 illegal access event 1: DMA0 illegal access event pending
21	SYSCFGIAF	SYS CFG illegal access event flag bit 0: no SYS CFG illegal access event 1: SYS CFG illegal access event pending
20	PMUIAF	PMU illegal access event flag bit 0: no PMU illegal access event 1: PMU illegal access event pending
19	RTCIAF	RTC illegal access event flag bit 0: no RTC illegal access event 1: RTC illegal access event pending
18:17	Reserved	Must be kept at reset value
16	SDIOIAF	SDIO illegal access event flag bit

		0: no SDIO illegal access event 1: SDIO illegal access event pending
15	PKCAUIAF	PKCAU illegal access event flag bit 0: no PKCAU illegal access event 1: PKCAU illegal access event pending
14	TRNGIAF	TRNG illegal access event flag bit 0: no TRNG illegal access event 1: TRNG illegal access event pending
13	HAUIAF	HAU illegal access event flag bit 0: no HAU illegal access event 1: HAU illegal access event pending
12	CAUIAF	CAU illegal access event flag bit 0: no CAU illegal access event 1: CAU illegal access event pending
11	ADCIAF	ADC illegal access event flag bit 0: no ADC illegal access event 1: ADC illegal access event pending
10	ICACHEIAF	ICACHE illegal access event flag bit 0: no ICACHE illegal access event 1: ICACHE illegal access event pending
9	TSIIAF	TSI illegal access event flag bit 0: no TSI illegal access event 1: TSI illegal access event pending
8	CRCIAF	CRC illegal access event flag bit 0: no CRC illegal access event 1: CRC illegal access event pending
7	HPDFIAF	HPDF illegal access event flag bit 0: no HPDF illegal access event 1: HPDF illegal access event pending
6:5	Reserved	Must be kept at reset value
4	TIMER16IAF	TIMER16 illegal access event flag bit 0: no TIMER16 illegal access event 1: TIMER16 illegal access event pending
3	TIMER15IAF	TIMER15 illegal access event flag bit 0: no TIMER15 illegal access event 1: TIMER15 illegal access event pending
2	Reserved	Must be kept at reset value

1	USART0IAF	USART0 illegal access event flag bit 0: no USART0 illegal access event 1: USART0 illegal access event pending
0	Reserved	Must be kept at reset value

9.9.6. TZIAC status register 2 (TZPCU_TZIAC_STAT2)

Address offset: 0x018

Reset value: 0x0000 0000

Secure access only.

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
WIFIIAF	DCIIAF	I2S1_AD	WIFI_RFI	QSPI_FL	SQPI_PS	QSPI_FL	SQPI_PS	EFUSEIA							Reserved
r	r	r	r	r	r	r	r	r	r	r					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TZBMPC 3_REGIA	SRAM3IA F	TZBMPC 2_REGIA	SRAM2IA F	TZBMPC 1_REGIA	SRAM1IA F	TZBMPC 0_REGIA	SRAM0IA F	Reserved	TZIACIAF	TZSPCIA F	
				r	r	r	r	r	r	r	r		r	r	

Bits	Fields	Descriptions
31	WIFIIAF	Wi-Fi illegal access event flag bit 0: No Wi-Fi illegal access event 1: Wi-Fi illegal access event pending
30	DCIIAF	DCI illegal access event flag bit 0: no DCI illegal access event 1: DCI illegal access event pending
29	I2S1_ADDIAF	I2S1_ADD illegal access event flag bit 0: no I2S1_ADD illegal access event 1: I2S1_ADD illegal access event pending
28	WIFI_RFIASF	EIFI RF illegal access event flag bit 0: no EIFI RF illegal access event 1: EIFI RF illegal access event pending
27	QSPI_FLASHREGIA F	QSPI FLASHREG illegal access event flag bit 0: no QSPI FLASHREG illegal access event 1: QSPI FLASHREG illegal access event pending

26	SQPI_PSRA MREGIA F	SQSPI PSRAMREG illegal access event flag bit 0: no SQSPI PSRAMREG illegal access event 1: SQSPI PSRAMREG illegal access event pending
25	QSPI_FLASHIAF	QSPI FLASH illegal access event flag bit 0: no QSPI FLASH illegal access event 1: QSPI FLASH illegal access event pending
24	SQPI_PSRA MIA F	SQPI PSRAM illegal access event flag bit 0: no SQPI PSRAM illegal access event 1: SQPI PSRAM illegal access event pending
23	EFUSEIAF	EFUSE illegal access event flag bit 0: no EFUSE illegal access event 1: EFUSE illegal access event pending
22:12	Reserved	Must be kept at reset value
11	TZBMPC3_REGIA F	TZBMPC3 REG illegal access event flag bit 0: no TZBMPC3 REG illegal access event 1: TZBMPC3 REG illegal access event pending
10	SRAM3IAF	SRAM3 illegal access event flag bit 0: no SRAM3 illegal access event 1: SRAM3 illegal access event pending
9	TZBMPC2_REGIA F	TZBMPC2 REG illegal access event flag bit 0: no TZBMPC2 REG illegal access event 1: TZBMPC2 REG illegal access event pending
8	SRAM2IAF	SRAM2 illegal access event flag bit 0: no SRAM2 illegal access event 1: SRAM2 illegal access event pending
7	TZBMPC1_REGIA F	TZBMPC2 REG illegal access event flag bit 0: no TZBMPC2 REG illegal access event 1: TZBMPC2 REG illegal access event pending
6	SRAM1IAF	SRAM1 illegal access event flag bit 0: no SRAM1 illegal access event 1: SRAM1 illegal access event pending
5	TZBMPC0_REGIA F	TZBMPC1 REG illegal access event flag bit 0: no TZBMPC1 REG illegal access event 1: TZBMPC1 REG illegal access event pending
4	SRAM0IAF	SRAM0 illegal access event flag bit 0: no SRAM0 illegal access event 1: SRAM0 illegal access event pending

3:2	Reserved	Must be kept at reset value
1	TZIACIAF	TZIAC illegal access event flag bit 0: no TZIAC illegal access event 1: TZIAC illegal access event pending
0	TZSPCIAF	TZSPC illegal access event flag bit 0: no TZSPC illegal access event 1: TZSPC illegal access event pending

9.9.7. TZIAC flag clear register 0 (TZPCU_TZIAC_STATC0)

Address offset: 0x020

Reset value: 0x0000 0000

Secure access only.

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
SPI0IAFC	TIMER0IAFC	Reserved		USBFSIAFC	Reserved										
w	w			w											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1IAFC	I2C0IAFC	Reserved		USART2IAFC	USART1IAFC	Reserved	SPI1IAFC	FWDGTIAFC	WWDGTIAC	Reserved	TIMER5IAFC	TIMER4IAFC	TIMER3IAFC	TIMER2IAFC	TIMER1IAFC
w	w			w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	SPI0IAFC	SPI0 illegal access flag clear bit This bit is set by softw are. 0: No action 1: Clear SPI0 illegal access flag
30	TIMER0IAFC	TIMER0 illegal access flag clear bit This bit is set by softw are. 0: No action 1: Clear TIMER0 illegal access flag
29:27	Reserved	Must be kept at reset value
26	USBFSIAFC	USBFS illegal access flag clear bit This bit is set by softw are. 0: No action 1: Clear USBFS illegal access flag
25:16	Reserved	Must be kept at reset value

15	I2C1IAFC	I2C1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear I2C1 illegal access flag
14	I2C0IAFC	I2C0 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear I2C0 illegal access flag
13:12	Reserved	Must be kept at reset value
11	USART2IAFC	USART2 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear USART2 illegal access flag
10	USART1IAFC	USART1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear USART1 illegal access flag
9	Reserved	Must be kept at reset value
8	SPI1IAFC	SPI1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SPI1 illegal access flag
7	FWDGTIA FC	FWDGT illegal access flag clear bit This bit is set by software. 0: No action 1: Clear FWDGT illegal access flag
6	WWDGTIAFC	WWDGT illegal access flag clear bit This bit is set by software. 0: No action 1: Clear WWDGT illegal access flag
5	Reserved	Must be kept at reset value
4	TIMER5IA FC	TIMER5 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TIMER5 illegal access flag
3	TIMER4IA FC	TIMER4 illegal access flag clear bit This bit is set by software. 0: No action

		1: Clear TIMER4 illegal access flag
2	TIMER3IA FC	TIMER3 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TIMER3 illegal access flag
1	TIMER2IA FC	TIMER2 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TIMER2 illegal access flag
0	TIMER1IA FC	TIMER1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TIMER1 illegal access flag

9.9.8. TZIAC flag clear register 1 (TZPCU_TZIAC_STATC1)

Address offset: 0x024

Reset value: 0x0000 0000

Secure access only.

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		EXTIIAFC	FMCIAFC	FLASHIA FC	RCUIAFC	Reserved	DMA1IAF C	DMA0IAF C	SYSCFGI AFC	PMUIAFC	RTClAFC	Reserved	SDIOIAF C		
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
PKCAUIA FC	TRNGIAF C	HAUIAFC	CAUIAFC	ADClAFC	ICACHEI AFC	TSIIAFC	CRCIAFC	HPDFIAF C	Reserved	TIMER16I AFC	TIMER15I AFC	Reserved	USART10I AFC	Reserved	
w	w	w	w	w	w	w	w	w		w	w	w	w		

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	EXTIIAFC	EXTI illegal access flag clear bit This bit is set by software. 0: No action 1: Clear EXTI illegal access flag
27	FMCIAFC	FMC illegal access flag clear bit This bit is set by software. 0: No action

		1: Clear FMC illegal access flag
26	FLASHIAFC	FLASH illegal access flag clear bit This bit is set by software. 0: No action 1: Clear FLASH illegal access flag
25	RCUIAFC	RCU illegal access flag clear bit This bit is set by software. 0: No action 1: Clear RCU illegal access flag
24	Reserved	Must be kept at reset value
23	DMA1IAFC	DMA1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear DMA1 illegal access flag
22	DMA0IAFC	DMA0 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear DMA0 illegal access flag
21	SYSCFGIAFC	SYSCFG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SYSCFG illegal access flag
20	PMUIAFC	PMU illegal access flag clear bit This bit is set by software. 0: No action 1: Clear PMU illegal access flag
19	RTCAIAFC	RTC illegal access flag clear bit This bit is set by software. 0: No action 1: Clear RTC illegal access flag
18:17	Reserved	Must be kept at reset value
16	SDIOIAFC	SDIO illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SDIO illegal access flag
15	PKCAUIAFC	PKCAU illegal access flag clear bit This bit is set by software. 0: No action

		1: Clear PKCAU illegal access flag
14	TRNGIAFC	<p>TRNG illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear TRNG illegal access flag</p>
13	HAUIAFC	<p>HAU illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear HAU illegal access flag</p>
12	CAUIAFC	<p>CAU illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear CAU illegal access flag</p>
11	ADCIADC	<p>ADC illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear ADC illegal access flag</p>
10	ICACHEIAFC	<p>ICACHE illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear ICACHE illegal access flag</p>
9	TSIIAFC	<p>TSI illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear TSI illegal access flag</p>
8	CRCIAFC	<p>CRC illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear CRC illegal access flag</p>
7	HPDFIAFC	<p>HPDF illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear HPDF illegal access flag</p>
6:5	Reserved	Must be kept at reset value
4	TIMER16IAFC	<p>TIMER16 illegal access flag clear bit</p> <p>This bit is set by software.</p> <p>0: No action</p> <p>1: Clear TIMER16 illegal access flag</p>

3	TIMER15IAFC	TIMER15 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TIMER15 illegal access flag
2	Reserved	Must be kept at reset value
1	USART0IAFC	USART0 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear USART0 illegal access flag
0	Reserved	Must be kept at reset value

9.9.9. TZIAC flag clear register 2 (TZPCU_TZIAC_STATC2)

Address offset: 0x028

Reset value: 0x0000 0000

Secure access only.

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
WIFIIAFC	DCIIAFC	I2S1_AD	WIFI_RFI	QSPI_FL	SQPI_PS	QSPI_FL	SQPI_PS	EFUSEIA							Reserved
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
Reserved				TZBMPC 3_REGIA FC	SRAM3IA FC	TZBMPC 2_REGIA FC	SRAM2IA FC	TZBMPC 1_REGIA FC	SRAM1IA FC	TZBMPC 0_REGIA FC	SRAM0IA FC	Reserved	TZIACIAF C	TZSPCIA FC	
				w	w	w	w	w	w	w	w		w	w	

Bits	Fields	Descriptions
31	WIFIIAFC	Wi-Fi illegal access flag clear bit This bit is set by software. 0: No action 1: Clear Wi-Fi illegal access flag
30	DCIIAFC	DCI illegal access flag clear bit This bit is set by software. 0: No action 1: Clear DCI illegal access flag
29	I2S1_ADDIAFC	I2S1_ADD illegal access flag clear bit This bit is set by software. 0: No action

		1: Clear I2S1_ADD illegal access flag
28	WIFI_RFIAFC	Wi-Fi RF illegal access flag clear bit This bit is set by software. 0: No action 1: Clear Wi-Fi RF illegal access flag
27	QSPI_FLASHREGIA FC	QSPI FLASHREG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear QSPI FLASHREG illegal access flag
26	SQPI_PSRAMREGIA FC	SQPI PSRAMREG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SQPI PSRAMREG illegal access flag
25	QSPI_FLASHIAFC	QSPI FLASH illegal access flag clear bit This bit is set by software. 0: No action 1: Clear QSPI FLASH illegal access flag
24	SQPI_PSRAMIAFC	SQPI PSRAM illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SQPI PSRAM illegal access flag
23	EFUSEIAFC	EFUSE illegal access flag clear bit This bit is set by software. 0: No action 1: Clear EFUSE illegal access flag
22:12	Reserved	Must be kept at reset value
11	TZBMP3_REGIAFC	TZBMP3 REG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZBMP3 REG illegal access flag
10	SRAM3IAFC	SRAM3 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SRAM3 illegal access flag
9	TZBMP2_REGIAFC	TZBMP2 REG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZBMP3 REG illegal access flag

8	SRAM2IAFC	SRAM2 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SRAM2 illegal access flag
7	TZBMP1_REGIAFC	TZBMP1 REG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZBMP1 REG illegal access flag
6	SRAM1IAFC	SRAM1 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SRAM1 illegal access flag
5	TZBMP0_REGIAFC	TZBMP0 REG illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZBMP0 REG illegal access flag
4	SRAM0IAFC	SRAM0 illegal access flag clear bit This bit is set by software. 0: No action 1: Clear SRAM0 illegal access flag
3:2	Reserved	Must be kept at reset value
1	TZIACIAFC	TZIAC illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZIAC illegal access flag
0	TZSPCIAFC	TZSPC illegal access flag clear bit This bit is set by software. 0: No action 1: Clear TZSPC illegal access flag

10. CRC calculation 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.

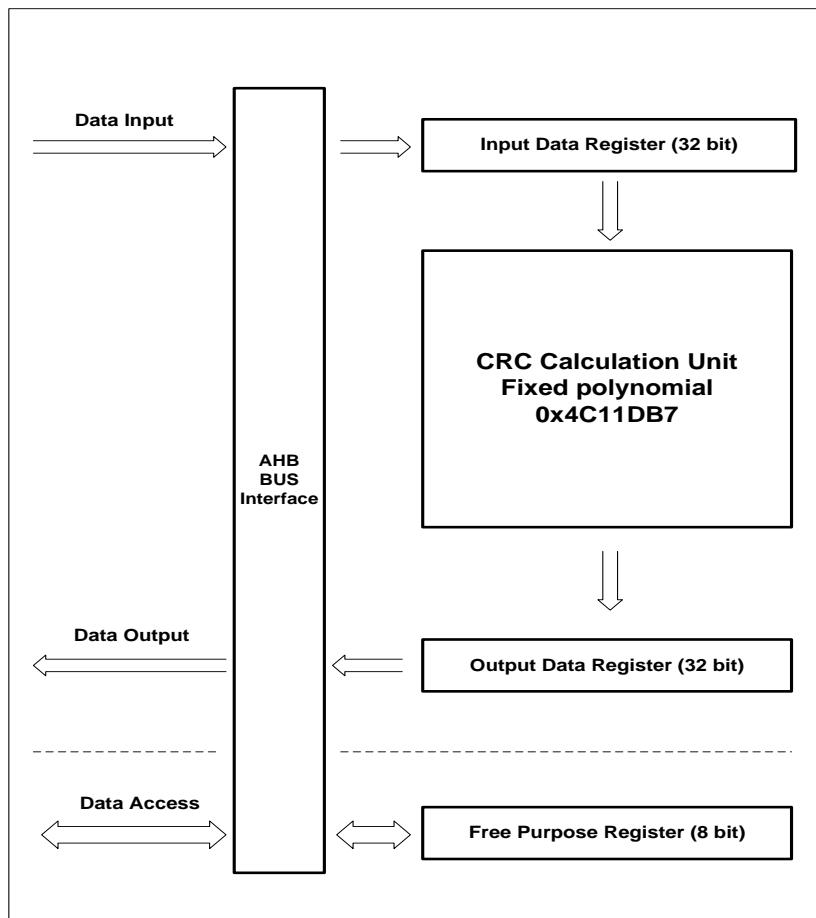
This CRC calculation unit can be used to calculate 32 bit CRC code with fixed polynomial.

10.2. Characteristics

- 32-bit data input and 32-bit data output. Calculation period is 4 AHB clock cycles for 32-bit input data size from data entered to the calculation result available.
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices.
- Fixed polynomial: 0x4C11DB7

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$
 This 32-bit CRC polynomial is a common polynomial used in Ethernet.

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 software 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 AHB clock cycles for 32-bit data size, during this period AHB will not be hanged because of the existence of the 32-bit input buffer.
- This module supplies an 8-bit free register CRC_FDATA.
 - CRC_FDATA is unrelated to the CRC calculation, any value you write in will be read out at anytime.

10.4. Register definition

CRC secure access base address: 0x5002 3000
 CRC non-secure access 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).

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

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. Written 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).

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

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 take 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).

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

rs

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	RST	Set this bit can reset the CRC_DATA register to the value of 0xFFFFFFFF then automatically cleared itself to 0 by hardware. This bit will take no effect to CRC_FDATA. Software writes and reads.

11. True random number generator (TRNG)

11.1. Overview

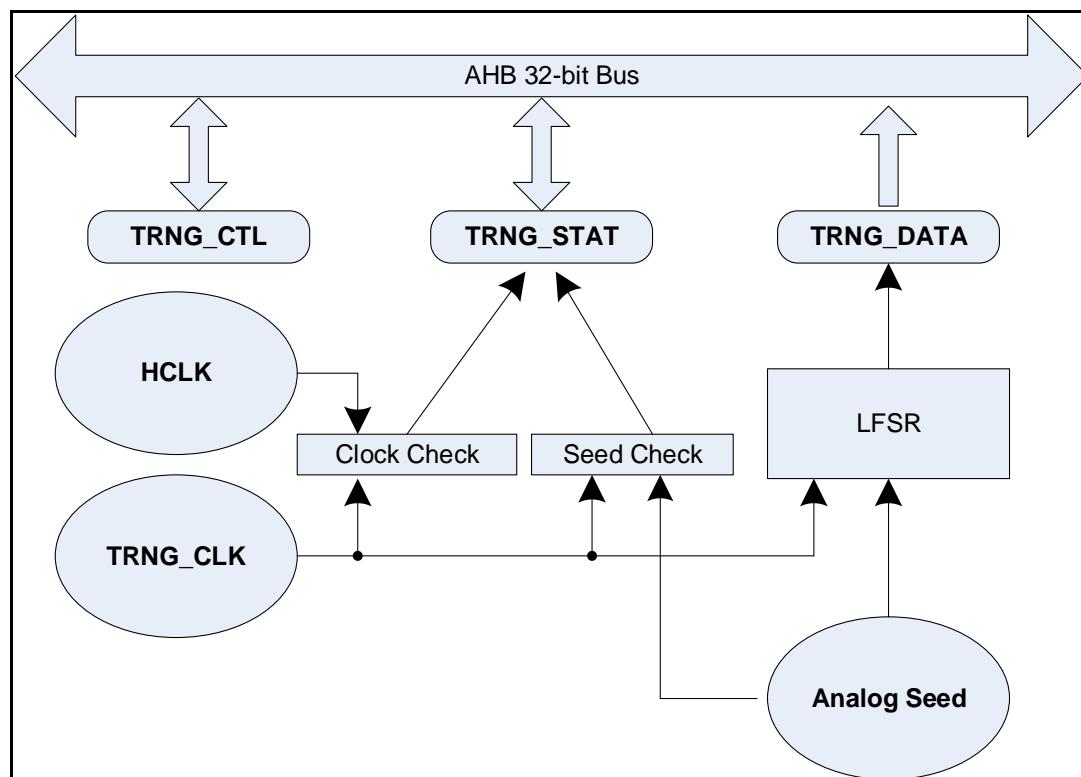
The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise.

11.2. Characteristics

- About 40 periods of TRNG_CLK are needed between two consecutive random numbers
- Disable TRNG module will significantly reduce the chip power consumption
- 32-bit random value seed is generated from analog noise, so the random number is a true random number.

11.3. Function overview

Figure 11-1. TRNG block diagram



The random number seed comes from analog circuit. This analog seed is then plugged into a linear feedback shift register (LFSR), where a 32-bit width random number is generated.

The analog seed is generated by several ring oscillators. The LFSR is driven by a configurable

TRNG_CLK (refer to [Reset and clock unit \(RCU\)](#) chapter), so that the quality of the generated random number depends on TRNG_CLK exclusively, no matter what HCLK frequency was set or not.

The 32-bit value of LFSR will transfer into TRNG_DATA register after a sufficient number of seeds have been sent to the LFSR.

At the same time, the analog seed and TRNG_CLK clock are monitored. When an analog seed error or a clock error occurs, the corresponding status bit in TRNG_STAT will be set and an interrupt will generate if the IE bit in TRNG_CTL is set.

11.3.1. Operation flow

The following steps are recommended for using TRNG block:

- 1). Enable the interrupt as necessary, so that when a random number or an error occurs, an interrupt will be generated.
- 2). Enable the TRNGEN bit.
- 3). When an interrupt occurs, check the status register TRNG_STAT, if SEIF=0, CEIF=0 and DRDY=1, then the random value in the data register could be read.

As required by the FIPS PUB 140-2, the first random data in data register should be saved but not be used. Every subsequent new random data should be compared to the previously random data. The data can only be used if it is not equal to the previously one.

11.3.2. Error flags

(1) Clock error

When the TRNG_CLK frequency is lower than the 1/16 of HCLK, the CECS and CEIF bit will be set. In this case, the application should check TRNG_CLK and HCLK frequency configurations and then clear CEIF bit. Clock error will not impact the previous random data.

(2) Seed error

When the analog seed is not changed or always changing during 64 TRNG_CLK periods, the SECS and SEIF bit will be set. In this case, the random data in data register should not be used. The application needs to clear the SEIF bit, then clear and set TRNGEN bit for restarting the TRNG.

11.4. Register definition

TRNG secure base address: 0x5C06 0800

TRNG non-secure base address: 0x4C06 0800

11.4.1. Control register (TRNG_CTL)

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

rw rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	IE	Interrupt enabled bit. This bit controls the generation of an interrupt when DRDY, SEIF or CEIF was set. 0: disable TRNG interrupt 1: enable TRNG interrupt
2	TRNGEN	TRNG enabled bit. 0: disable TRNG module (reduce power consuming) 1: enable TRNG module
1:0	Reserved	Must be kept at reset value.

11.4.2. Status register (TRNG_STAT)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SEIF	CEIF	Reserved	SECS	CECS	DRDY

rc_w0 rc_w0 r r r

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	SEIF	<p>Seed error interrupt flag</p> <p>This bit will be set if more than 64 consecutive same bit or more than 32 consecutive 01(or 10) changing are detected.</p> <p>0: No fault detected</p> <p>1: Seed error has been detected. The bit is cleared by writing 0.</p>
5	CEIF	<p>Clock error interrupt flag</p> <p>This bit will be set if TRNG_CLK frequency is lower than 1/16 HCLK frequency.</p> <p>0: No fault detected</p> <p>1: Clock error has been detected. The bit is cleared by writing 0.</p>
4:3	Reserved	Must be kept at reset value.
2	SECS	<p>Seed error current status</p> <p>0: Seed error is not detected at current time. In case of SEIF=1 and SECS=0, it means seed error has been detected before but now is recovered.</p> <p>1: Seed error is detected at current time if more than 64 consecutive same bits or more than 32 consecutive 01(or 10) changing are detected</p>
1	CECS	<p>Clock error current status</p> <p>0: Clock error is not detected at current time. In case of CEIF=1 and CECS=0, it means clock error has been detected before but now is recovered.</p> <p>1: Clock error is detected at current time. TRNG_CLK frequency is lower than 1/16 HCLK frequency.</p>
0	DRDY	<p>Random data ready status bit. This bit is cleared by reading the TRNG_DATA register and set when a new random number is generated.</p> <p>0: The content of TRNG data register is not available.</p> <p>1: The content of TRNG data register is available.</p>

11.4.3. Data register (TRNG_DATA)

Address offset: 0x08

Reset value: 0x0000 0000

Application must make sure DRDY is set before reading this register

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
TRNDATA[31:16]															

r

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

TRNDATA[15:0]

r

Bits	Fields	Descriptions
31:0	TRNDATA[31:0]	32-bit random data

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 MCU, thereby increasing system performance by off-loading the MCU from copying large amounts of data and avoiding frequent interrupts to serve peripherals needing more data or having available data.

Two AHB master interfaces and eight four-word depth 32-bit width FIFOs are presented in each DMA controller, which achieves a high DMA transmission performance. There are 16 independent channels in the DMA controller (8 for DMA0 and 8 for DMA1). Each channel is assigned a specific or multiple target peripheral devices for memory access request management. Two arbiters respectively for memory and peripheral are implemented inside to handle the priority among DMA requests.

Both the DMA controller and the Cortex-M33 core implement data access through the system bus. An arbitration mechanism is implemented to solve the competition between these two masters. When the same peripheral is targeted, the MCU access will be suspended for some specific bus cycles. A round-robin scheduling algorithm is utilized in the bus matrix to guarantee at least half the bandwidth to the MCU.

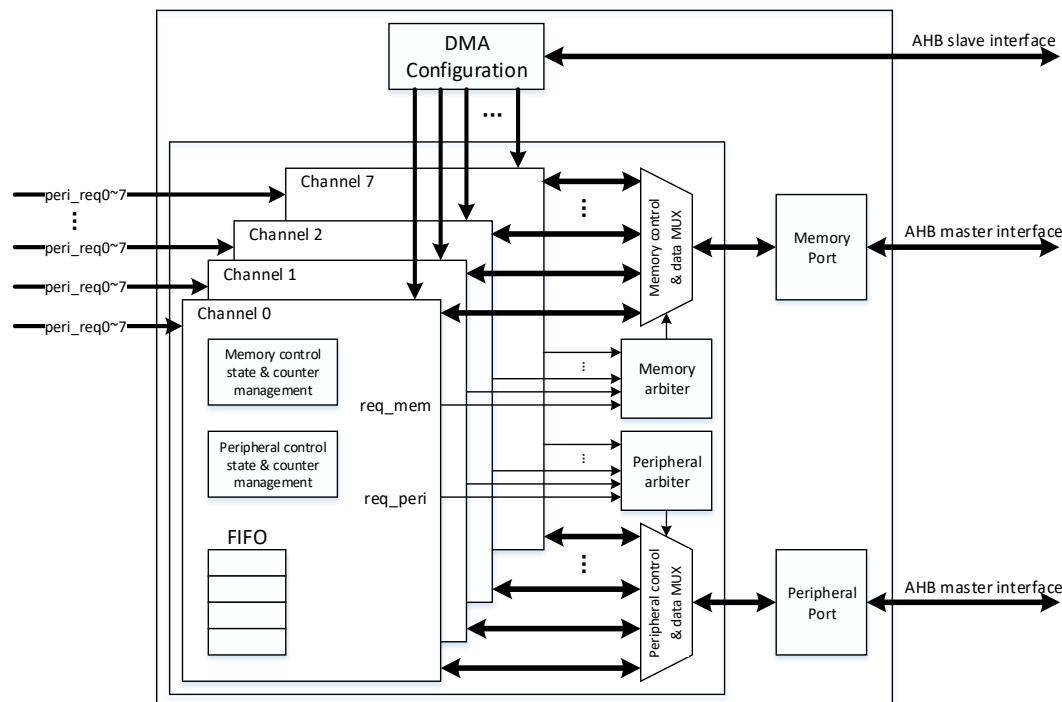
12.2. Characteristics

- Two AHB master interface for transferring data, and one AHB slave interface for programming DMA.
- 16 channels (8 for DMA0 and 8 for DMA1), up to 8 peripherals per channel with fixed hardware peripheral requests.
- Support independent single, 4, 8, 16-beat incrementing burst memory and peripheral transfer.
- Support switch-buffer transmission between peripheral and memory.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 7 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 three transfer modes:
 - Read from memory and write to peripheral.
 - Read from peripheral and write to memory.
 - Read from memory and write to memory (only for DMA1).

- Both DMA and peripheral can be configured as flow controller:
 - DMA: Programmable length of data to be transferred, max to 65535.
 - Peripheral: The last request signal given to DMA from peripheral determines the end of transfer.
- Support two data processing modes by use of the four-word depth 32-bit width FIFOs:
 - Multi-data mode: Pack/Unpack data when memory transfer width are different from peripheral transfer width.
 - Single-data mode: Read data from source when FIFO is empty and write data to destination when one data has been pushed into FIFO.
- One separate interrupt per channel with five 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 access through two AHB master interfaces respectively for memory access and peripheral access.
- Two arbiters inside to manage multiple peripheral requests coming at the same time.
- Channel data management to control data packing/unpacking and counting.

12.4. Function overview

The DMA controller transfers data from one address to another without CPU intervention. It supports multiple data sizes, burst types, address generation algorithm, priority levels and several transfer modes to allow for flexible application by configuring the corresponding bits in DMA registers. All the DMA registers can be 32-bit configured through AHB slave interface.

Three transfer modes are supported, including peripheral-to-memory, memory-to-peripheral and memory-to-memory, which is determined by the TM bits in the DMA_CHxCTL register, as listed in [Table 12-1. Transfer mode](#).

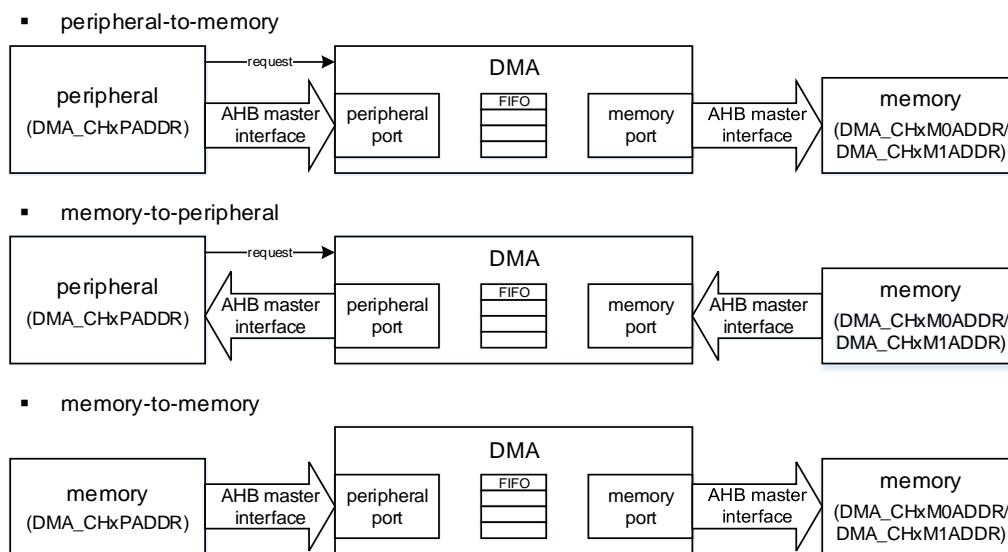
Table 12-1. Transfer mode

Transfer mode	TM[1:0]	Source	Destination
Peripheral to memory	00	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR1
Memory to peripheral	01	DMA_CHxM0ADDR/ DMA_CHxM1ADDR	DMA_CHxPADDR
Memory to memory	10	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR

Note:

1. The MBS bit in DMA_CHxCTL register determines which is selected as the memory buffer address in DMA_CHxM0ADDR and DMA_CHxM1ADDR register. For more information, refer to section [Switch-buffer mode](#).
2. The TM bits in DMA_CHxCTL register are forbidden to configure to 0b11, or the channel will be automatically disabled.

Figure 12-2. Data stream for three transfer modes



As shown in [Figure 12-2. Data stream for three transfer modes](#), Two AHB master

interfaces are implemented in each DMA respectively for memory and peripheral.

- Memory to peripheral: read data from memory through AHB master interface for memory, and write data to peripheral through AHB master interface for peripheral.
- Peripheral to memory: read data from peripheral through AHB master interface for peripheral, and write data to memory through AHB master interface for memory.
- Memory to memory: read data from memory through AHB master interface for peripheral, and write data to another memory through AHB master interface for memory.

12.4.1. Secure & privileged

Security

The DMA controller is compliant with the TrustZone-M hardware architecture, partitioning all its resources so that they exist in one of the two worlds: the secure world and the nonsecure world, at any given time.

A secure software is able to access any resource/register, whatever secure or non-secure.

A non-secure software is restricted to access any non-secure resource/register.

Any channel is in a secure or non-secure state, as securely configured by the DMA_CHxSCTL.SECM secure register bit.

When a channel x is configured in secure mode, the following access controls rules are applied:

- A non-secure read access to a register field of this channel is forced to return 0, except for both the secure state and the privileged state of this channel x (SECM and PRIV bits of the DMA_CHxSCTL register) which are readable by a non-secure software.
- A non-secure write access to a register field of this channel has no impact.

When a channel is configured in secure mode, a secure software can separately configure as secure or non-secure the AHB DMA master transfer from the source (by the DMA_CHxSCTL.SSEC register bit), and as secure or non-secure the AHB DMA master transfer to the destination (by the DMA_CHxSCTL.DSEC register bit).

The DMA controller also generates a security illegal access pulse, on an illegal non-secure software access to a secure DMA register or register field. This event is routed to the TrustZone interrupt controller.

The illegal access pulse is generated in the configurations described below:

- If the channel x is in secure state (SECM bit of the DMA_CHxSCTL register set), the illegal access pulse is generated on one of the following accesses:
 - a non-secure write access to a dedicated register of this channel x (DMA_CHxCTL, DMA_CHxCNT, DMA_CHxPADDR, DMA_CHxM0ADDR, DMA_CHxM1ADDR, DMA_CHxFCTL and DMA_CHxSCTL).
 - a non-secure read access to a dedicated register of this channel x, except the DMA_CHxSCTL register (DMA_CHxCTL, DMA_CHxCNT, DMA_CHxPADDR, DMA_CHxM0ADDR DMA_CHxM1ADDR and DMA_CHxFCTL).
- If the channel x is in non-secure state (SECM bit of the DMA_CHxSCTL register

cleared), the illegal access pulse is generated on a non-secure write access to the DMA_CHxSCTL register which attempts to write 1 into any of the secure configuration bits SECM, DSEC, SSEC.

When the software is switching from a secure state to a non-secure state (after the secure transfer is completed), the secure software must disable the channel by a 32-bit write at the DMA_CHxCTL address before switching. This operation is needed for the two below reasons:

- a non-secure software cannot do so.
- the EN bit of the DMA_CHxCTL register must be cleared before the (non-secure) software can reprogram the DMA_CHxCTL for a next transfer.

The dma_secm signal is asserted when any channels secure mode bit (SECMx) is set.

Privileged / unprivileged mode

The DMA controller performs AHB master transfers, with a privileged or unprivileged access mode, at a channel level.

When a channel x is configured in privileged mode, the following access controls rules are applied:

- An unprivileged read access to a register field of this channel is forced to return 0, except for both the privileged state and the secure state of this channel x (PRIV and SECM bits of the DMA_CHxSCTL register) which are readable by an unprivileged software.
- An unprivileged write access to a register field of this channel has no impact.

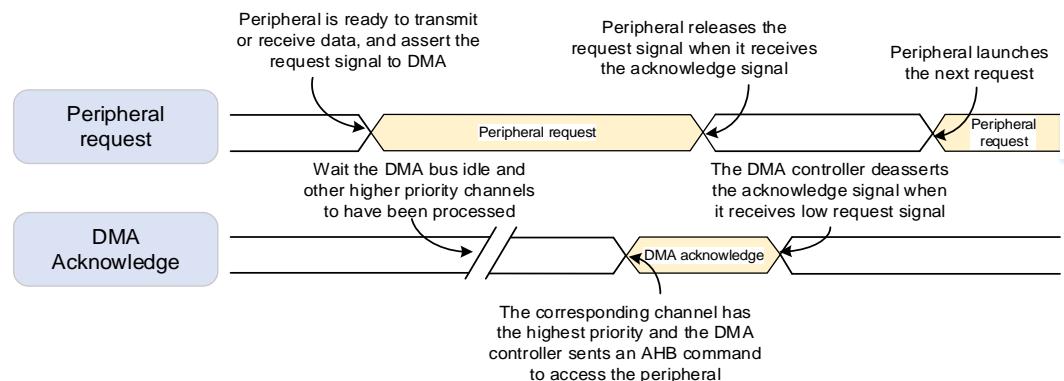
12.4.2. Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and a 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 12-3. Handshake mechanism**](#) shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 12-3. Handshake mechanism



Each DMA has 8 channels, with fixed multiple peripheral requests. The PERIEN bits in the DMA_CHxCTL register determine which peripheral request signal connects to each DMA channel. The peripheral requests mapping of DMA0 is listed in [Table 12-2. Peripheral requests to DMA0](#), and the peripheral requests mapping of DMA1 is listed in [Table 12-3. Peripheral requests to DMA1](#).

As listed in the [Table 12-2. Peripheral requests to DMA0](#) and [Table 12-3. Peripheral requests to DMA1](#), a peripheral request can be connected to two different DMA channels. It is forbidden to simultaneously enable these two DMA channels with selecting the same peripheral request. For example, in DMA0, I2C0_RX is connected to channel 0 and channel 5. When the PERIEN bits in the DMA_CH0CTL register are configured to 0b001 and the PERIEN bits in the DMA_CH5CTL register are configured to 0b001, enable these two channels and responding the request from I2C0 at the same time will cause transmission error.

Table 12-2. Peripheral requests to DMA0

Channel		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
PERIEN[2:0]	000	•	•	•	SPI1_RX	SPI1_TX	•	•	•
	001	I2C0_RX	•	•	•	•	I2C0_RX	I2C0_TX	I2C0_TX
	010	TIMER3_CH0	•	•	TIMER3_OH1	I2S1_ADDTX	•	TIMER3_UTIMER3_C	H2
	011	•	TIMER1_UP	•	I2S1_ADD_RX	•	TIMER1_OH0	TIMER1_UTIMER1_U	P
	100	•	USART2_RX	•	USART2_TX	•	USART1_RX	USART1_TX	•
	101	•	•	TIMER2_CH3	•	TIMER2_OH0	TIMER2_OH1	•	TIMER2_CH2
	110	TIMER4_CH	TIMER4_CH	TIMER4_CH	TIMER4_O	TIMER4_O	•	TIMER4_UTIMER4_U	•

		2 TIMER4_UP	CH3 TIMER4_T G	0	H3 TIMER4_T G	H1		P	
	111	•	TIMER5_UP	I2C1_RX	I2C1_RX	USART2_TX	•	•	I2C1_TX

Table 12-3. Peripheral requests to DMA1

Channel	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
PERIEN[2:0]	000	ADC0	•	•	•	ADC0	•	TIMER0_C H0 TIMER0_C H1 TIMER0_C H2
	001	•	DCI	•	•	•	•	DCI
	010	•	•	•	•	•	CAU_OUT	CAU_IN
	011	SPI0_RX	•	SPI0_RX	SPI0_TX	•	SPI0_TX	•
	100	•	•	USART0_RX	SDIO	•	USART0_RX	SDIO
	101	QUADSPI	QUADSPI	TIMER15_C H0 TIMER15_U P	TIMER16_CH0 TIMER16_UP	•	•	TIMER15_CH0 CH0 TIMER15_UP TIMER16_UP
	110	TIMER0_TG	TIMER0_C H0	TIMER0_CH 1	TIMER0_C H0	TIMER0_C G MT	TIMER0_U P	TIMER0_C H2
	111	•	HPDF_FL T0	HPDF_FLT1	•	•	•	•

12.4.3. Data process

Arbitration

Two arbiters are implemented in each DMA respectively for memory and peripheral port. When two or more requests are received at the same time, 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.

Transfer width, burst and counter

Transfer width

PWIDTH and MWIDTH in the DMA_CHxCTL register indicate the data width of a peripheral and memory transfer separately. The DMA supports 8-bit, 16-bit and 32-bit transfer width. In multi-data mode, if PWIDTH is not equal to MWIDTH, the DMA can automatically packs/unpacks data to achieve an integrated and correct data transfer operation. In single-data mode, MWIDTH is automatically locked as PWIDTH by hardware immediately after enable the DMA channel.

Transfer burst type

PBURST and MBURST in the DMA_CHxCTL register indicate the burst type of a peripheral and memory transfer separately. The DMA supports single burst, 4-beat, 8-beat and 16-beat incrementing burst for peripheral port and memory port. In single-data mode, only single burst type is supported and PBURST and MBURST are automatically locked as '00' by hardware immediately after enable the DMA channel.

In peripheral-to-memory or memory-to-peripheral mode, if PBURST is different from '00', DMA responses a increasing burst transfer of 4, 8, 16-beat based on the PBURST bits for each peripheral request. If the remaining bytes number of data item to be transferred is less than the bytes number needed for a burst transfer, the remaining data items are transferred in single transaction.

AMBA protocol specifies that bursts must not cross a 1kB address boundary, or a transfer error will be responded to the master. In each DMA, the peripheral burst transfer crossing a 1kB address boundary is decomposed to 4, 8 or 16 single transactions depend on the PBURST bits, as the same as the memory burst transfer.

Transfer counter

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. If the peripheral is configured as the flow controller, the CNT bits are forced to '0xFFFF' immediately after enabling the channel whatever the CNT bits are. During the transmission, the CNT bits indicate the remaining number of data items to be transferred.

The CNT bits are related to peripheral transfer width, the number of data bytes to be transferred is the CNT bits multiplied by the byte number of the peripheral transfer width. For example, if the PWIDTH bits are equal to '11', and the number of data bytes to be transferred is CNT \times 4. The CNT bits is decreased by 1 when a single or a beat of the burst peripheral transfer (the source memory transfer in the memory-to-memory mode) has been completed even if the transfer mode is peripheral-to-memory or memory-to-memory.

When configuring the CNT bits, the following rules must be respected to guarantee a good DMA operation:

If the circular mode is disabled by clearing the CMEN bit in the DMA_CHxCTL register, the rules to configure the CNT bits in the DMA_CHxCNT register based on the transfer width are listed in the [Table 12-4. CNT configuration](#).

The number of data bytes must be an integer multiple of the memory transfer width to guarantee an integrated single memory transfer.

Note: The number of data bytes does not need to be an integer multiple of the bytes number of a memory burst transfer or a peripheral burst number if the PBURST or/and MBURST bits are not equal to '00'. The remaining data not enough for a burst transfer are transferred can be divided into single transaction automatically.

Table 12-4. CNT configuration

PWIDTH	MWIDTH	CNT
8-bit	16-bit	Multiple of 2
8-bit	32-bit	Multiple of 4
16-bit	32-bit	Multiple of 2
Others		Any value

1. If the circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register. The number of data bytes must be an integer multiple of the byte number of a peripheral burst transfer and a memory burst transfer to guarantee an integrated memory and peripheral burst transfer:
 - a) $\text{CNT} / \text{PBURST_beats}$ must be an integer.
 - b) $(\text{CNT} \times \text{PWIDTH_bytes}) / (\text{MBURST_beats} \times \text{MWIDTH_bytes})$ must be an integer.
 - ① PWIDTH_bytes is the byte number of the peripheral transfer width, 1 for 8-bit, 2 for 16-bit and 4 for 32-bit.
 - ② PBURST_beats is the beat number of a peripheral burst transfer, 1 for single burst, 4 for INCR4, 8 for INCR8 and 16 for INCR16.
 - ③ MWIDTH_bytes is the byte number of the peripheral transfer width, 1 for 8-bit, 2 for 16-bit and 4 for 32-bit.
 - ④ MBURST_beats is the beat number of a peripheral burst transfer, 1 for single burst, 4 for INCR4, 8 for INCR8 and 16 for INCR16.

For example:

1. If PWIDTH is 16-bit, PBURST is INCR4, MWIDTH is 8-bit and MBURST is INCR16, $\text{CNT}/4$ and $(\text{CNT} \times 2) / (1 \times 16)$ must be an integer, so the CNT bits must be configured to the multiple of 8.
2. If PWIDTH is 8-bit, PBURST is INCR16, MWIDTH is 16-bit and MBURST is INCR4, $\text{CNT}/16$ and $(\text{CNT} \times 1) / (2 \times 4)$ must be an integer, so the CNT bits must be configured to the multiple of 16.

Note: when the switch-buffer mode is enabled by setting the SBMEN bit in the DMA_CHxCTL register, the circular mode is enabled automatically by hardware, and the above rules must also be respected.

FIFO

A four-word depth 32-bit FIFO is implemented as a data buffer for each DMA channel. Data reading from the source address is stored in the FIFO temporarily and transmitted to the destination through the destination port. Two data processing modes are supported depend on the FIFO configuration, including single-data mode and multi-data mode. When the transfer mode is memory-to-memory, only multi-data mode is supported to implement the DMA data processing.

Multi-data mode

The multi-data mode is selected by configuring the MDMEN bit in the DMA_CHxFCTL register to '1'.

In this mode, the DMA responds the source request when there is enough FIFO space for a source transfer, pushing the data reading from the source address into the FIFO. If the destination is a peripheral, the DMA responds the peripheral request when there is enough FIFO data for a peripheral burst transfer. If the memory is configured as the destination, the FIFO counter critical value configured in the FCCV bits of the DMA_CHxFCTL register controls the memory data processing. Only when the FIFO counter is reached the critical value, the data in the FIFO are entirely popped and written into the memory address.

To guarantee a good DMA behavior, the FIFO counter critical value (FCCV bits in the DMA_CHxFCTL register) must be an integer multiple of a memory burst transfer to ensure there is enough data for memory burst transfers. The configuration rules of the FIFO counter critical value depending on memory transfer width and memory burst types are listed in [Table 12-5. FIFO counter critical value configuration rules](#).

Table 12-5. FIFO counter critical value configuration rules

MWIDTH	MBURST	FIFO counter critical value			
		1-word	2-word	3-word	4-word
8-bit	single	4 single transactions	8 single transactions	12 single transactions	16 single transactions
	INCR4	1 burst transaction	2 burst transactions	3 burst transactions	4 burst transactions
	INCR8	ERROR	1 burst transaction	ERROR	2 burst transactions
	INCR16	ERROR	ERROR	ERROR	1 burst transaction
16-bit	single	2 single transactions	4 single transactions	6 single transactions	8 single transactions
	INCR4	ERROR	1 burst transaction	ERROR	2 burst transactions
	INCR8	ERROR	ERROR	ERROR	1 burst transaction
	INCR16	ERROR	ERROR	ERROR	ERROR

	single	1 single transaction	2 single transactions	3 single transactions	4 single transactions
32-bit	INCR4	ERROR	ERROR	ERROR	1 burst transactions
	INCR8	ERROR	ERROR	ERROR	ERROR
	INCR16	ERROR	ERROR	ERROR	ERROR

Note: When the transfer mode is peripheral-to-memory, if the $PBURST_beats \times PWIDTH_bytes = 16$, the FIFO counter critical value must not be equal to '10'. When receiving a peripheral request, DMA initiates a peripheral burst transfer to entirely fill the FIFO. Then DMA launches memory burst transfers to pop three words from the FIFO depending on the FIFO counter critical value and a word is still remained in the FIFO. There is no enough space for a peripheral burst transfer and the FIFO counter critical value is not reached, which make DMA transfer frozen.

Single-data mode

The single-data mode is selected by configuring the MDMEN bit in the DMA_CHxFCTL register to '0'. In this mode, only single transfer is supported to implement the DMA data access, and the FIFO counter critical value configured in the FCCV bits of the DMA_CHxFCTL register has no meaning.

In single-data mode, DMA responds the source request only when the FIFO is empty, pushing the data reading from the source address into the FIFO whatever the source transfer width is. When the FIFO is not empty, DMA responds the destination request, popping the data from the FIFO and writing it to the destination address.

Pack/Unpack

In single-data mode, the MWIDTH bits are equal to the PWIDTH bits by force, data packing/unpacking is not needed.

In multi-data mode, the independent PWIDTH and MWIDTH bits configuration are supported for flexible DMA transfer. When the PWIDTH bits and the MWIDTH bits are not equal, DMA reading access and writing access are executed in different transfer width, and DMA packs/unpacks the data automatically. In DMA transfer operation, only little-endian addressing for both memory and peripheral is supported.

Suppose the CNT bits are 16, the PWIDTH bits are equal to '00', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the [Figure 12-4. Data packing/unpacking when PWIDTH = '00'.](#)

Figure 12-4. Data packing/unpacking when PWIDTH = '00'

- PAIF = 0, MWIDTH = 8-bit

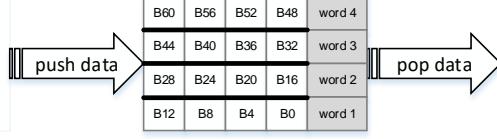
```
read 0xB0[7:0] @0x0 read 0xB8[7:0] @0x8
read 0xB1[7:0] @0x1 read 0xB9[7:0] @0x9
read 0xB2[7:0] @0x2 read 0xB10[7:0] @0xA
read 0xB3[7:0] @0x3 read 0xB11[7:0] @0xB
read 0xB4[7:0] @0x4 read 0xB12[7:0] @0xC
read 0xB5[7:0] @0x5 read 0xB13[7:0] @0xD
read 0xB6[7:0] @0x6 read 0xB14[7:0] @0xE
read 0xB7[7:0] @0x7 read 0xB15[7:0] @0xF
```



```
write 0xB0[7:0] @0x0 write 0xB8[7:0] @0x8
write 0xB1[7:0] @0x1 write 0xB9[7:0] @0x9
write 0xB2[7:0] @0x2 write 0xB10[7:0] @0xA
write 0xB3[7:0] @0x3 write 0xB11[7:0] @0xB
write 0xB4[7:0] @0x4 write 0xB12[7:0] @0xC
write 0xB5[7:0] @0x5 write 0xB13[7:0] @0xD
write 0xB6[7:0] @0x6 write 0xB14[7:0] @0xE
write 0xB7[7:0] @0x7 write 0xB15[7:0] @0xF
```

- PAIF = 1, MWIDTH = 16-bit

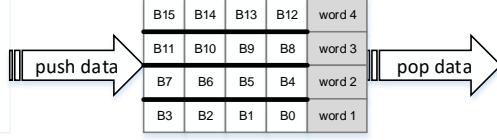
```
read 0xB0[7:0] @0x0 read 0xB32[7:0] @0x20
read 0xB4[7:0] @0x4 read 0xB36[7:0] @0x24
read 0xB8[7:0] @0x8 read 0xB40[7:0] @0x28
read 0xB12[7:0] @0xC read 0xB44[7:0] @0x2C
read 0xB16[7:0] @0x10 read 0xB48[7:0] @0x30
read 0xB20[7:0] @0x14 read 0xB52[7:0] @0x34
read 0xB24[7:0] @0x18 read 0xB56[7:0] @0x38
read 0xB28[7:0] @0x1C read 0xB60[7:0] @0x3C
```



```
write 0xB4B0[15:0] @0x0
write 0xB12B8[15:0] @0x2
write 0xB20B16[15:0] @0x4
write 0xB28B24[15:0] @0x6
write 0xB36B32[15:0] @0x8
write 0xB44B40[15:0] @0xA
write 0xB52B48[15:0] @0xC
write 0xB60B56[15:0] @0xE
```

- PAIF = 0, MWIDTH = 32-bit

```
read 0xB0[7:0] @0x0 read 0xB8[7:0] @0x8
read 0xB1[7:0] @0x1 read 0xB9[7:0] @0x9
read 0xB2[7:0] @0x2 read 0xB10[7:0] @0xA
read 0xB3[7:0] @0x3 read 0xB11[7:0] @0xB
read 0xB4[7:0] @0x4 read 0xB12[7:0] @0xC
read 0xB5[7:0] @0x5 read 0xB13[7:0] @0xD
read 0xB6[7:0] @0x6 read 0xB14[7:0] @0xE
read 0xB7[7:0] @0x7 read 0xB15[7:0] @0xF
```



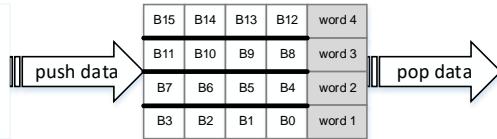
```
write 0xB3B2B1B0[31:0] @0x0
write 0xB7B6B5B4[31:0] @0x4
write 0xB11B10B9B8[31:0] @0x8
write 0xB15B14B13B12[31:0] @0xC
```

- Suppose the CNT bits are 8, the PWIDHT bits are equal to '01', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the [Figure 12-5. Data packing/unpacking when PWIDHT = '01'](#).

Figure 12-5. Data packing/unpacking when PWIDHT = '01'

- PAIF = 0, MWIDTH = 8-bit

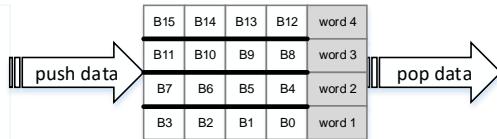
```
read 0xB1B0[15:0] @0x0
read 0xB3B2[15:0] @0x2
read 0xB5B4[15:0] @0x4
read 0xB7B6[15:0] @0x6
read 0xB9B8[15:0] @0x8
read 0xB11B10[15:0] @0xA
read 0xB13B12[15:0] @0xC
read 0xB15B14[15:0] @0xE
```



```
write 0xB0[7:0] @0x0 write 0xB8[7:0] @0x8
write 0xB1[7:0] @0x1 write 0xB9[7:0] @0x9
write 0xB2[7:0] @0x2 write 0xB10[7:0] @0xA
write 0xB3[7:0] @0x3 write 0xB11[7:0] @0xB
write 0xB4[7:0] @0x4 write 0xB12[7:0] @0xC
write 0xB5[7:0] @0x5 write 0xB13[7:0] @0xD
write 0xB6[7:0] @0x6 write 0xB14[7:0] @0xE
write 0xB7[7:0] @0x7 write 0xB15[7:0] @0xF
```

- PAIF = 0, MWIDTH = 16-bit

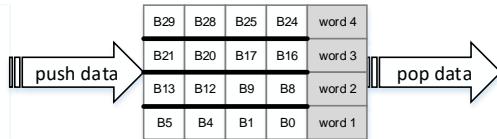
```
read 0xB1B0[15:0] @0x0
read 0xB3B2[15:0] @0x2
read 0xB5B4[15:0] @0x4
read 0xB7B6[15:0] @0x6
read 0xB9B8[15:0] @0x8
read 0xB11B10[15:0] @0xA
read 0xB13B12[15:0] @0xC
read 0xB15B14[15:0] @0xE
```



```
write 0xB1B0[15:0] @0x0
write 0xB3B2[15:0] @0x2
write 0xB5B4[15:0] @0x4
write 0xB7B6[15:0] @0x6
write 0xB9B8[15:0] @0x8
write 0xB11B10[15:0] @0xA
write 0xB13B12[15:0] @0xC
write 0xB15B14[15:0] @0xE
```

- PAIF = 1, MWIDTH = 32-bit

```
read 0xB1B0[15:0] @0x0
read 0xB5B4[15:0] @0x4
read 0xB9B8[15:0] @0x8
read 0xB13B12[15:0] @0xC
read 0xB17B16[15:0] @0x10
read 0xB21B20[15:0] @0x14
read 0xB25B24[15:0] @0x18
read 0xB29B28[15:0] @0x1C
```



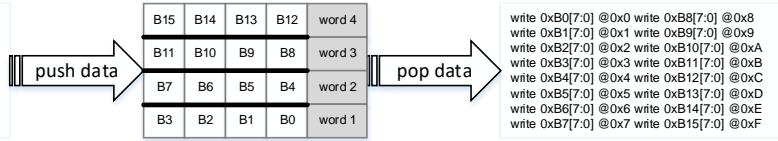
```
write 0xB5B4B1B0[31:0] @0x0
write 0xB13B12B9B8[31:0] @0x4
write 0xB21B20B17B16[31:0] @0x8
write 0xB29B28B25B24[31:0] @0xC
```

- Suppose DMA_CHxCNT is 4, the PWIDHT bits are equal to '10', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the [Figure 12-6. Data packing/unpacking when PWIDHT = '10'](#).

Figure 12-6. Data packing/unpacking when PWIDTH = ‘10’

- PAIF = 1, MWIDTH = 8-bit

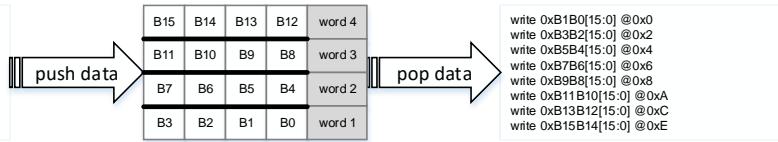
```
read 0xB3B2B1B0[31:0] @0x0
read 0xB7B6B5B4[31:0] @0x4
read 0xB11B10B9B8[31:0] @0x8
read 0xB15B14B13B12[31:0] @0xC
```



```
write 0xB0[7:0] @0x0 write 0xB8[7:0] @0x8
write 0xB1[7:0] @0x1 write 0xB9[7:0] @0x9
write 0xB2[7:0] @0x2 write 0xB10[7:0] @0xA
write 0xB3[7:0] @0x3 write 0xB11[7:0] @0xB
write 0xB4[7:0] @0x4 write 0xB12[7:0] @0xC
write 0xB5[7:0] @0x5 write 0xB13[7:0] @0xD
write 0xB6[7:0] @0x6 write 0xB14[7:0] @0xE
write 0xB7[7:0] @0x7 write 0xB15[7:0] @0xF
```

- PAIF = 0, MWIDTH = 16-bit

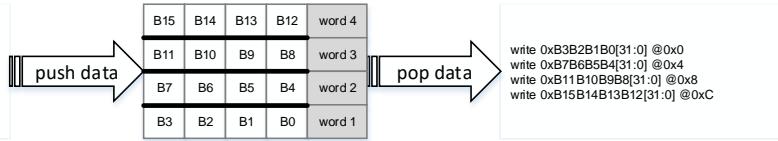
```
read 0xB3B2B1B0[31:0] @0x0
read 0xB7B6B5B4[31:0] @0x4
read 0xB11B10B9B8[31:0] @0x8
read 0xB15B14B13B12[31:0] @0xC
```



```
write 0xB1B0[15:0] @0x0
write 0xB3B2[15:0] @0x2
write 0xB5B4[15:0] @0x4
write 0xB7B6[15:0] @0x6
write 0xB9B8[15:0] @0x8
write 0xB11B10[15:0] @0xA
write 0xB13B12[15:0] @0xC
write 0xB15B14[15:0] @0xE
```

- PAIF = 0, MWIDTH = 32-bit

```
read 0xB3B2B1B0[31:0] @0x0
read 0xB7B6B5B4[31:0] @0x4
read 0xB11B10B9B8[31:0] @0x8
read 0xB15B14B13B12[31:0] @0xC
```



```
write 0xB3B2B1B0[31:0] @0x0
write 0xB7B6B5B4[31:0] @0x4
write 0xB11B10B9B8[31:0] @0x8
write 0xB15B14B13B12[31:0] @0xC
```

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_CHxM0ADDR, and DMA_CHxM1ADDR).

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. In Multi-data mode with PBURST in the DMA_CHxCTL register different from ‘00’, if PAIF in the DMA_CHxCTL register is enable, the next peripheral address increment is fixed to 4, and has nothing to do with the peripheral transfer data width. The PAIF has no meaning to the memory address generation.

Note: If PAIF in the DMA_CHxCTL register is enable, the peripheral base address configured in the DMA_CHxPADDR register must be 32-bit alignment.

12.4.5. Circular mode

Circular mode is implemented to handle continue peripheral requests. The CMEN bit in the DMA_CHxCTL register is used to enable/disable the circular mode. Circular mode is available only when DMA controls the transfer flow. When the peripheral is selected as the transfer flow controller by setting the TFCS, the circular mode is automatically disabled immediately after the channel is enabled.

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 a transfer error is detected or the CHEN bit in the

DMA_CHxCTL register is cleared.

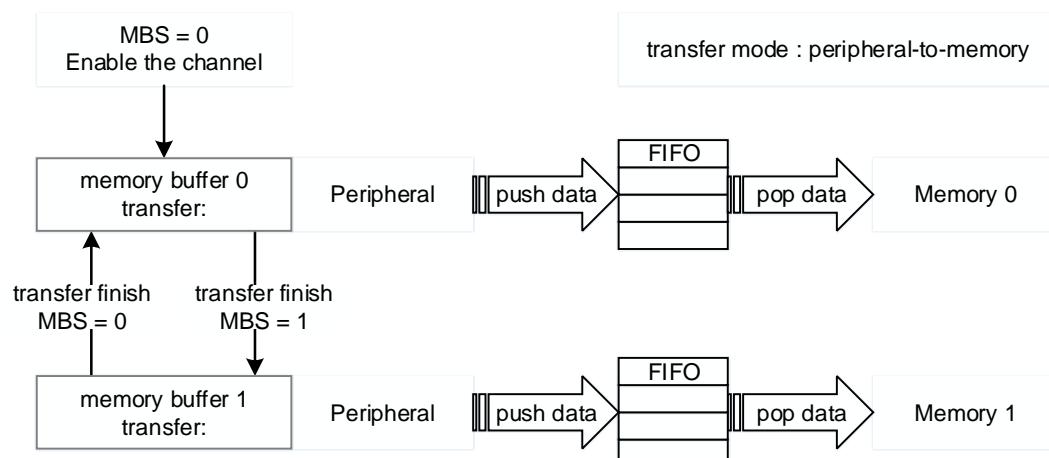
12.4.6. Switch-buffer mode

Similar to circular mode, switch-buffer mode is also implemented to handle continues peripheral requests. The SBMEN bit in the DMA_CHxCTL register is used to enable/disable the switch-buffer mode. When the switch-buffer mode is enabled, the circular mode is automatically enabled immediately after the channel is enabled. Switch-buffer mode is only available when the transfer mode is peripheral-to-memory or memory-to-peripheral. When the transfer mode is memory-to-memory, the switch-buffer mode is automatically disabled immediately after the channel is enabled.

Switch-buffer mode is supported with two memory buffers and the base address of the two memory buffers are separately configured in the DMA_CHxM0ADDR and DMA_CHxM1ADDR register. In switch-buffer mode, the DMA memory pointer switches from the current memory buffer to another at the end of every DMA transfer. During the DMA transmission, the memory buffer not being processed by DMA can be accessed by other AHB masters. In switch-buffer mode, the base address of the memory buffer not accessed by DMA can be updated even if the channel is enabled.

The MBS bit in the DMA_CHxCTL register is configured to select which memory buffer is accessed by DMA at the first DMA transfer before the channel is enabled. In switch-buffer mode, this bit switches automatically between '0' and '1' at the end of every DMA transfer, and can be used as a flag indicating the current memory buffer accessed by DMA during the transmission. The DMA operation of switch-buffer mode are shown in [Figure 12-7. DMA operation of switch-buffer mode](#).

Figure 12-7. DMA operation of switch-buffer mode



12.4.7. Transfer flow controller

The transfer flow controller controls the number of data items to be transferred. The TFCS bit in the DMA_CHxCTL register determines which of DMA and peripheral is selected to control

the transfer flow.

- DMA as transfer flow controller: The CNT bits in the DMA_CHxCNT register determine the number of data items to be transferred. The CNT bits must be configured before the channel is enabled.
- Peripheral as transfer flow controller: The CNT bits configured in the DMA_CHxCNT register before the channel is enabled have no meaning and these bits are forced to '0xFFFF' immediately after the channel is enabled. The peripheral determines when to finish the DMA transfer by informing a last request signal to DMA.

Note: When the transfer mode is memory-to-memory, the transfer flow controller is fixed to be DMA whatever the TFCS bit is configured to.

12.4.8. Transfer operation

Three transfer modes are supported to implement the data transfer, including peripheral-to-memory, memory-to-peripheral and memory-to-memory. Memory and peripheral can be configured as source and destination relatively.

Memory transfer

- Peripheral-to-memory mode:
 - In single-data mode, when the FIFO is not empty, DMA initiates a single memory transfer and writes data into the corresponding memory address.
 - In multi-data mode, when the FIFO counter reaches the critical value, DMA starts single or burst memory transfers to entirely fetch the FIFO data and write to the memory.
- Memory-to-peripheral mode:
 - In single-data mode, when the channel is enabled, DMA starts a single memory transfer and pushes the reading data into the FIFO immediately. During the transmission, the memory transfer is initiated only when the FIFO is empty.
 - In multi-data mode, when the channel is enabled, DMA starts several single or burst transfers to fill up the FIFO whether the peripheral request is asserted or not. During the transmission, the memory transfer is initiated once when there is enough space for it in the FIFO.
- Memory-to-memory mode: Only the multi-data mode is supported. When the FIFO counter reaches the critical value, DMA starts single or burst memory transfers to entirely fetch the FIFO data and write to the memory.

Peripheral transfer

- Peripheral-to-memory mode: When receiving a peripheral request and there is enough space in the FIFO for a peripheral transfer, DMA starts a peripheral transfer and pushes the reading data into the FIFO.
- Memory-to-peripheral mode: When receiving a peripheral request and there is enough

data in the FIFO for a peripheral transfer, DMA starts a peripheral transfers to fetch the FIFO data and write to the peripheral.

- Memory-to-memory mode: Only the multi-data mode is supported. When the channel is enabled, DMA starts several peripheral transfers to fill up the FIFO. During the transmission, the peripheral transfer is initiated once when there is enough space for it in the FIFO.

12.4.9. Transfer finish

The DMA transfer is finished automatically and the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register is set when one of the following situations occurs:

- Transfer completion
- Software clear
- Error detection

Transfer completion

When enabled, the DMA begins to transfer data between peripheral and memory. After the pre-programmed number of data items has been transferred successfully, the DMA transfer is completed and the CHEN bit is automatically cleared in the DMA_CHxCTL register.

- Peripheral-to-memory mode: If DMA is the transfer flow controller, when the CNT bits reach to zero and the contents of the FIFO have been entirely transferred into the memory, an end of transfer is generated. If peripheral is the transfer flow controller, the DMA transfer is completed when the last peripheral request has been responded and the contents of the FIFO have been entirely transferred into the memory.
- Memory-to-peripheral mode: If DMA is the transfer flow controller, when the CNT bits in the DMA_CHxCNT register reach to zero, an end of transfer is achieved. If peripheral is the transfer flow controller, the DMA transfer is completed when the last peripheral request has been responded.
- Memory-to-memory: only DMA can be the transfer flow controller. When the CNT bits reach to zero and the contents of the FIFO have been entirely transferred into the memory, an end of transfer is generated.

Software clear

The DMA transfer can be stopped by clearing the CHEN bit in the DMA_CHxCTL register by software. After the software cleared operation, the CHEN bit is still read as 1 to indicate that there are memory or peripheral transfers still active or the remaining data in the FIFO need to be transferred.

- Peripheral-to-memory: After the software cleared operation, the peripheral transfer is stopped when the current single or burst transfer is completed. To ensure that the data had been read from peripheral can be entirely transferred into the memory, the memory transfer continues to be active until the FIFO is empty. If the remaining byte number in

the FIFO is not enough for a burst memory transfer, these data items are transferred in single transaction. If the remaining byte number is less than the memory transfer width, these data items are still written in memory transfer width with MSBs filled with zero. The software can read the CNT bits to calculate the number of valid data items in the memory. After the contents of the FIFO has been entirely transferred into the memory, the CHEN bit is cleared automatically by hardware and the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register is set.

- Memory-to-peripheral: After the software cleared operation, the DMA transfer is stopped when the current memory and peripheral transfer are completed. Then the CHEN bit is cleared and the FTFIFx bit is set.
- Memory-to-memory: The same as the peripheral-to-memory mode with the source memory transfer is implemented through the peripheral port.

Error detection

Three types error can disable the DMA transfer:

- FIFO error: When a wrong FIFO configuration is detected, the DMA channel is disabled immediately without starting any transfers. In this situation, the FTFIFx is not asserted. For more information about the FIFO error, refer to section [Error](#).
- Bus error: When the memory or peripheral port attempts to access an address beyond the access scope, a bus error is detected and the DMA transfer is stopped immediately without setting the FTFIFx. If this error is aroused by the peripheral port, the CNT bits are still decreased by 1. For more information about the bus error, refer to section [Error](#).
- Register access error: In switch-buffer mode, an access error is detected when a write command is active on the memory base address register which is being accessed by DMA. When this error occurs, the DMA operation is the same as it after the CHEN bit software cleared. For more information about the register access error, refer to section [Error](#).

12.4.10. 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 or wait the current DMA transfer finished. When the CHEN bit is read as '0', configuring and starting a new DMA transfer is allowed.
2. Clear the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register, or a new DMA transfer can not be re-enabled.
3. Configure the TM bits in the DMA_CHxCTL register to set the transfer mode.
4. Configure the PERIEN bits in the DMA_CHxCTL register to select the target peripheral. If the transfer mode is memory-to-memory, the PERIEN bits have no meaning and this

step can be skipped.

5. Configure the memory and peripheral burst types, the target memory buffer, switch-buffer mode, priority of the channel, memory and peripheral transfer width, memory and peripheral address generation algorithm, circular mode, the transfer flow controller in the DMA_CHxCTL register.
6. Configure multi-data mode, and the FCCV bits to set the FIFO counter critical value if multi-data mode is enabled in the DMA_CHxFCTL register.
7. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer access error interrupt, single-data mode exception interrupt in the DMA_CHxCTL register and the enable bit for FIFO error and exception interrupt in the DMA_CHxFCTL register.
8. Configure the DMA_CHxPADDR register for setting the peripheral base address.
9. If the switch-buffer mode is enabled, configure the DMA_CHxM0ADDR and DMA_CHxM1ADDR register for setting the memory base address. If only one memory buffer is to be used, configure the DMA_CHxM0ADDR or DMA_CHxM1ADDR corresponding with the MBS bit in the DMA_CHxCTL register.
10. Configure the DMA_CHxCNT register to set the total transfer data number.
11. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the channel.

When restarting the suspended DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and ensure the DMA suspend operation has been completed. When the CHEN bit is read as '0', restarting the DMA transfer is allowed.
2. Clear the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register, or the DMA transfer can not be re-enabled.
3. Read the DMA_CHxCNT register to obtain the number of the remaining data items and calculate the number of the data items had already been transferred.
4. Configure the DMA_CHxPADDR register to update the peripheral address pointer.
5. Configure the DMA_CHxM0ADDR or the DMA_CHxM1ADDR register to update the memory address pointer.
6. Configure the DMA_CHxCNT with the number of the remaining data items.
7. Configure the CHEN bit with '1' in the DMA_CHxCTL register to restart the channel.

12.5. Interrupts

Each DMA channel has a dedicated interrupt. There are five interrupt events connected to each interrupt, including full transfer finish interrupt, half transfer finish interrupt, transfer

access error interrupt, single-data mode exception interrupt, and FIFO error and exception interrupt. A DMA channel interrupt may be produced when any interrupt event occurs on the channel.

Each interrupt event has a dedicated flag bit in the DMA_INTF0 or DMA_INTF1 register, a dedicated clear bit in the DMA_INTC0 and DMA_INTC1 register, and a dedicated enable bit in the DMA_CHxCTL and CHxFCTL register, as described in the [Table 12-6. DMA interrupt events](#).

Table 12-6. DMA interrupt events

Interrupt event	Flag bit	Enable bit	Clear bit
	DMA_INTF0 or DMA_INTF1	DMA_CHxCTL or DMA_CHxFCTL	DMA_INTC0 or DMA_INTC1
Full transfer finish	FTFIF	FTFIE	FTFIFC
Half transfer finish	HTFIF	HTFIE	HTFIFC
Transfer access error	TAEIF	TAEIE	TAEIFC
Single-data mode exception	SDEIF	SDEIE	SDEIFC
FIFO error and exception	FEEIF	FEEIE	FEEIFC

These five events can be divided into three types:

- Flag: Full transfer finish flag and half transfer finish flag
- Exception: Single-data mode exception and FIFO exception
- Error: Transfer access error and FIFO error

When the exception events occur, the DMA transmission is not affected and continues transferring normally. When the error events are detected, the DMA transmission is stopped. These three types of event are described in detail in the following sections.

12.5.1. Flag

Two flag events are supported, including full transfer finish flag and half transfer finish flag.

The full transfer finish flag is asserted, when one of the following situations occurs:

- The CNT bits reach to zero when DMA is the transfer flow controller.
- When peripheral is the transfer flow controller, the last request is responded completely and the contents of the FIFO are entirely written into the memory in peripheral-to-memory mode.
- When the channel is disabled by software before the end of the transfer, the current memory and peripheral is completed and the contents of the FIFO are entirely written into the memory in peripheral-to-memory or memory-to-memory mode.

- When the channel is disabled because of register access error before the end of the transfer, the current memory and peripheral is completed and the contents of the FIFO are entirely written into the memory in peripheral-to-memory or memory-to-memory mode.

When the full transfer finish flag is asserted and the enabled bit for the full transfer finish interrupt is set, an interrupt is generated.

The half transfer finish flag is asserted, only when DMA is the transfer flow controller and half of the CNT bits are transferred. If peripheral is the transfer flow controller, DMA does not know when half of data items has been transferred and the half transfer finish flag will stay zero.

When the half transfer finish flag is asserted and the enabled bit for the half transfer finish interrupt is set, an interrupt is generated.

12.5.2. Exception

Two exception events are supported, including single-data mode exception and FIFO exception. These exceptions have no effect on the DMA transmission.

Single-data mode exception

This exception can be detected only when the single-data mode is enabled and the transfer mode is peripheral-to-memory. When a peripheral request is valid and the FIFO is not empty, there are two or more data items stored in the FIFO after responding the peripheral request, which could be a problem for the subsequent processing of the data.

When the single-data mode exception is asserted and the enabled bit for the single-data mode exception interrupt is set, an interrupt is generated.

FIFO exception

When a FIFO underrun or a FIFO overrun condition occurs, the FIFO exception is asserted. This exception can be detected only when the transmission is between peripheral and memory.

In peripheral-to-memory mode, when a peripheral request is valid and there is not enough space in the FIFO for the single or burst peripheral transfer, a FIFO overrun condition is detected. This peripheral request is not responded until the FIFO space is enough, and the accuracy of the data transmission will not be destroyed.

In memory-to-peripheral mode, when a peripheral request is valid and there is not enough data in the FIFO for the single or burst peripheral, a FIFO underrun condition is detected. This peripheral request is not responded until the data number in the FIFO is enough, and the accuracy of the data transmission will not be destroyed.

When the FIFO exception is asserted and the enabled bit for the FIFO error and exception interrupt is set, an interrupt is generated.

12.5.3. Error

FIFO error and transfer access error (including the register access error and bus error) can be detected during the DMA transmission, and the transmission can be stopped when one of the errors occurs.

FIFO error

For a good DMA operation, when the multi-data mode is enabled, the right and wrong configurations of the FIFO counter critical value corresponding with the memory transfer width and memory burst types are listed in 错误!未找到引用源。

If a wrong configuration is detected after enable the channel, a FIFO error is generated and the channel is disabled immediately without starting any transfers.

When the FIFO error is asserted and the enabled bit for the FIFO error and exception interrupt is set, an interrupt is generated.

Register access error

The register access error is detected only when the switch-buffer is enabled. If the software attempts to update a memory address register currently accessed by the DMA controller, a register access error is detected. For example, when the memory 0 buffer is the current source or destination, a write access on the DMA_CHxM0ADDR register could produce a register access error. When a register access error occurs, the DMA transmission is stopped when the current memory and peripheral transfer are completed and the valid FIFO data are entirely drained into the memory if needed.

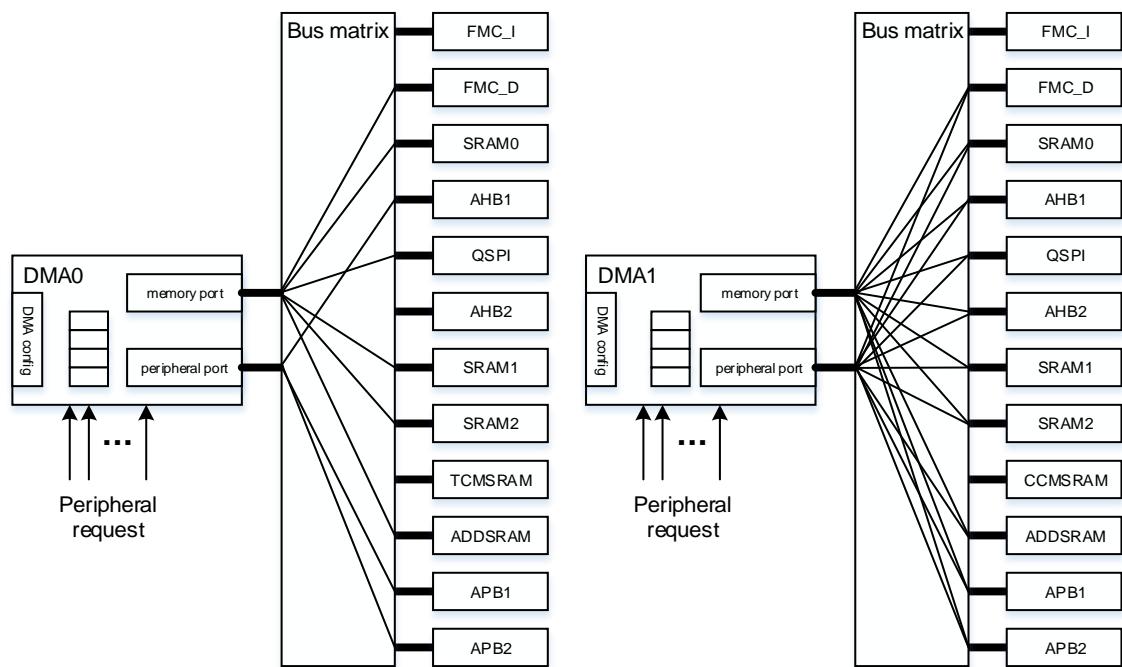
When the register access error is asserted and the enabled bit for the transfer access error and exception interrupt is set, an interrupt is generated.

Bus error

When the address accessed by the DMA controller is beyond the allowed area, a response error will be received and the channel is disabled immediately. The allowed and forbidden access region for DMA0 and DMA1 are shown in [Figure 12-8. System connection of DMA0 and DMA1](#). When the bus error is asserted and the enabled bit for the transfer access error

and exception interrupt is set, an interrupt is generated.

Figure 12-8. System connection of DMA0 and DMA1



12.6. Register definition

DMA0 secure access base address: 0x5002 6000

DMA0 non-Secure access base address: 0x4002 6000

DMA1 secure access base address: 0x5002 6400

DMA1 non-Secure access base address: 0x4002 6400

12.6.1. Interrupt flag register 0 (DMA_INTF0)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FTFIF3	HTFIF3	TAEIF3	SDEIF3	Reserved	FEEIF3	FTFIF2	HTFIF2	TAEIF2	SDEIF2	Reserved	FEEIF2
	r		r	r	r		r		r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				FTFIF1	HTFIF1	TAEIF1	SDEIF1	Reserved	FEEIF1	FTFIFO	HTFIFO	TAEIFO	SDEIFO	Reserved	FEEIFO
	r		r	r	r	r	r		r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFx	Full Transfer finish flag of channel x (x=0...3) Hardw are set and softw are cleared by configuring DMA_INTC0 register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
26/20/10/4	HTFIFx	Half transfer finish flag of channel x (x=0...3) Hardw are set and softw are cleared by configuring DMA_INTC0 register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/19/9/3	TAEIFx	Transfer access error flag of channel x (x=0...3) Hardw are set and softw are cleared by configuring DMA_INTC0 register. 0: Transfer access error has not occurred on channel x 1: Transfer access error has occurred on channel x
24/18/8/2	SDEIFx	Single data mode exception of channel x (x=0...3) Hardw are set and softw are cleared by configuring DMA_INTC0 register. 0: Single data mode exception has not occurred on channel x 1: Single data mode exception has occurred on channel x
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFx	FIFO error and exception of channel x (x=0...3)

Hardw are set and softw are cleared by configuring DMA_INTC0 register.

0: FIFO error or exception has not occurred on channel x

1: FIFO error or exception has occurred on channel x

12.6.2. Interrupt flag register 1 (DMA_INTF1)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		Reserved		FTFIF7	HTFIF7	TAEIF7	SDEIF7	Reserved	FEEIF7	FTFIF6	HTFIF6	TAEIF6	SDEIF6	Reserved	FEEIF6
				r	r	r	r		r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Reserved		FTFIF5	HTFIF5	TAEIF5	SDEIF5	Reserved	FEEIF5	FTFIF4	HTFIF4	TAEIF4	SDEIF4	Reserved	FEEIF4
				r	r	r	r		r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFx	Full Transfer finish flag of channel x (x=4...7) Hardw are set and softw are cleared by configuring DMA_INTC1 register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
26/20/10/4	HTFIFx	Half transfer finish flag of channel x (x=4...7) Hardw are set and softw are cleared by configuring DMA_INTC1 register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/19/9/3	TAEIFx	Transfer access error flag of channel x (x=4...7) Hardw are set and softw are cleared by configuring DMA_INTC1 register. 0: Transfer access error has not occurred on channel x 1: Transfer access error has occurred on channel x
24/18/8/2	SDEIFx	Single data mode exception of channel x (x=4...7) Hardw are set and softw are cleared by configuring DMA_INTC1 register. 0: Single data mode exception has not occurred on channel x 1: Single data mode exception has occurred on channel x
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFx	FIFO error and exception of channel x (x=4...7) Hardw are set and softw are cleared by configuring DMA_INTC1 register. 0: FIFO error or exception has not occurred on channel x 1: FIFO error or exception has occurred on channel x

12.6.3. Interrupt flag clear register 0 (DMA_INTC0)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		Reserved		FTFIFC3	HTFIFC3	TAEIFC3	SDEIFC3	Reserved	FEEIFC3	FTFIFC2	HTFIFC2	TAEIFC2	SDEIFC2	Reserved	FEEIFC2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Reserved		FTFIFC1	HTFIFC1	TAEIFC1	SDEIFC1	Reserved	FEEIFC1	FTFIFC0	HTFIFC0	TAEIFC0	SDEIFC0	Reserved	FEEIFC0
				w	w	w	w		w	w	w	w	w		w

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFCx	Clear bit for Full transfer finish flag of channel x (x=0...3) 0: No effect 1: Clear full transfer finish flag
26/20/10/4	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0...3) 0: No effect 1: Clear half transfer finish flag
25/19/9/3	TAEIFCx	Clear bit for transfer access error flag of channel x (x=0...3) 0: No effect 1: Clear transfer access error flag
24/18/8/2	SDEIFCx	Clear bit for single data mode exception of channel x (x=0...3) 0: No effect 1: Clear single data mode exception flag
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFCx	Clear bit for FIFO error and exception of channel x (x=0...3) 0: No effect 1: Clear FIFO error and exception flag

12.6.4. Interrupt flag clear register 1 (DMA_INTC1)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		Reserved		FTFIFC7	HTFIFC7	TAEIFC7	SDEIFC7	Reserved	FEEIFC7	FTFIFC6	HTFIFC6	TAEIFC6	SDEIFC6	Reserved	FEEIFC6
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Reserved		FTFIFC5	HTFIFC5	TAEIFC5	SDEIFC5	Reserved	FEEIFC5	FTFIFC4	HTFIFC4	TAEIFC4	SDEIFC4	Reserved	FEEIFC4
				w	w	w	w		w	w	w	w	w		w

w w w w w w w w w w

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=4...7) 0: No effect 1: Clear full transfer finish flag
26/20/10/4	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=4...7) 0: No effect 1: Clear half transfer finish flag
25/19/9/3	TAEIFCx	Clear bit for transfer access error flag of channel x (x=4...7) 0: No effect 1: Clear transfer access error flag
24/18/8/2	SDEIFCx	Clear bit for single data mode exception of channel x (x=4...7) 0: No effect 1: Clear single data mode exception flag
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFCx	Clear bit for FIFO error and exception of channel x (x=4...7) 0: No effect 1: Clear FIFO error and exception flag

12.6.5. Channel x control register (DMA_CHxCTL)

x = 0...7, where x is a channel number

Address offset: 0x10 + 0x18 × x

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PERIEN[2:0]			MBURST[1:0]		PBURST[1:0]		Reserved	MBS	SBMEN	PRIO[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
PAIF	MWIDTH[1:0]	PWIDTH[1:0]	MNAGA	PNAGA	CMEN	TM[1:0]		TFCS	FTFIE	HTFIE	TAEIE	SDEIE	CHEN		
	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.
27:25	PERIEN[2:0]	Peripheral enable Software set and clear. 000: Enable peripheral 0

		001: Enable peripheral 1 010: Enable peripheral 2 011: Enable peripheral 3 100: Enable peripheral 4 101: Enable peripheral 5 110: Enable peripheral 6 111: Enable peripheral 7 <p>These bits can NOT be written when CHEN is '1'.</p>
24:23	MBURST[1:0]	<p>Transfer burst type of memory</p> <p>Software set and clear.</p> <p>00: single burst 01: INCR4 (4-beat incrementing burst) 10: INCR8 (8-beat incrementing burst) 11: INCR16 (16-beat incrementing burst)</p> <p>These bits can NOT be written when CHEN is '1'.</p> <p>These bits are automatically locked as '00' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.</p>
22:21	PBURST[1:0]	<p>Transfer burst type of peripheral</p> <p>Software set and clear.</p> <p>00: single burst 01: INCR4 (4-beat incrementing burst) 10: INCR8 (8-beat incrementing burst) 11: INCR16 (16-beat incrementing burst)</p> <p>These bits can NOT be written when CHEN is '1'.</p> <p>These bits are automatically locked as '00' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.</p>
20	Reserved	Must be kept at reset value.
19	MBS	<p>Memory buffer select</p> <p>Hardware and software set, hardware and software clear.</p> <p>0: Memory 0 is selected as memory transfer area 1: Memory 1 is selected as memory transfer area</p> <p>This bit can NOT be written when CHEN is '1'.</p> <p>During the transmission, this bit can be set and cleared by hardware at the end of transfer to indicate which memory buffer is being accessed by DMA.</p>
18	SBMEN	<p>Switch-buffer mode enable</p> <p>Software set and clear.</p> <p>0: Disable switch-buffer mode 1: Enable switch-buffer mode</p> <p>This bit can NOT be written when CHEN is '1'.</p>
17:16	PRIO[1:0]	<p>Priority level</p> <p>Software set and clear.</p>

00: Low

01: Medium

10: High

11: Ultra high

These bits can NOT be written when CHEN is '1'.

15	PAIF	<p>Peripheral address increment fixed</p> <p>Software set and clear.</p> <p>0: The peripheral address increment is determined by PWIDTH</p> <p>1: The peripheral address increment is fixed to 4</p> <p>This bit can NOT be written when CHEN is '1'.</p> <p>During the transmission, when PNAGA is configured to '0', this bit has no effect.</p> <p>These bits are automatically locked as '0' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0' or PBURST are not equal to '00'.</p>
14:13	MWIDTH[1:0]	<p>Transfer width of memory</p> <p>Software set and clear.</p> <p>00: 8-bit</p> <p>01: 16-bit</p> <p>10: 32-bit</p> <p>11: Reserved</p> <p>These bits can NOT be written when CHEN is '1'.</p> <p>These bits are automatically locked as PWIDTH by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.</p>
12:11	PWIDTH[1:0]	<p>Transfer width of peripheral</p> <p>Software set and clear.</p> <p>00: 8-bit</p> <p>01: 16-bit</p> <p>10: 32-bit</p> <p>11: Reserved</p> <p>These bits can NOT be written when CHEN is '1'.</p>
10	MNAGA	<p>Next address generation algorithm of memory</p> <p>Software set and clear</p> <p>0: Fixed address mode</p> <p>1: Increasing address mode</p> <p>This bit can NOT be written when CHEN is '1'.</p>
9	PNAGA	<p>Next address generation algorithm of peripheral</p> <p>Software set and clear</p> <p>0: Fixed address mode</p> <p>1: Increasing address mode</p> <p>This bit can NOT be written when CHEN is '1'.</p>
8	CMEN	Circular mode enable

		Softw are set and clear. 0: Disable circular mode. 1: Enable circular mode This bit can NOT be written when CHEN is '1'. This bit is automatically locked as '0' by hardware immediately after enable CHEN if TFCS is configured to '1'. This bit is automatically locked as '1' by hardware immediately after enable CHEN if SBMEN is configured to '1'.
7:6	TM[1:0]	Transfer mode Softw are set and clear. 00: Read from peripheral and write to memory 01: Read from memory and write to peripheral 10: Read from memory and write to memory 11: Reserved These bits can NOT be written when CHEN is '1'.
5	TFCS	Transfer flow controller select Softw are set and clear. 0: DMA is selected as the transfer flow controller 1: Peripheral is selected as the transfer flow controller This bit can NOT be written when CHEN is '1'.
4	FTFIE	Enable bit for full transfer finish interrupt Softw are set and clear. 0:Disable full transfer finish interrupt 1:Enable full transfer finish interrupt
3	HTFIE	Enable bit for half transfer finish interrupt Softw are set and clear. 0: Disable half transfer finish interrupt 1: Enable half transfer finish interrupt
2	TAEIE	Enable bit for tranfer access error interrupt Softw are set and clear. 0: Disable tranfer access error interrupt 1: Enable tranfer access error interrupt
1	SDEIE	Enable bit for single data mode exception interrupt Softw are set and clear. 0: Disable single data mode exception interrupt 1: Enable single data mode exception interrupt
0	CHEN	Channel enable Softw are set, hardw are clear. 0: Disable channel 1: Enable channel

When this bit is asserted, the DMA transfer is started. This bit is automatically cleared when one of the following situations occurs:

When the transfer of channel is fully finished.

When a wrong FIFO configuration or a transfer access error is detected.

After a software clear operation, this bit is still read as 1 to indicate that there are memory or peripheral transfers still active until hardware has terminated all activity, at which point this bit is read as 0. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.

12.6.6. Channel x counter register (DMA_CHxCNT)

x = 0...7, where x is a channel number

Address offset: 0x14 + 0x18 × x

Reset value: 0x0000 0000

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	<p>Transfer counter</p> <p>These bits can NOT be written when CHEN in the DMA_CHxCTL register is '1'.</p> <p>These bits are related to PWIDTH. During the transmission, These bits signify the number of remaining data to be transferred. After each DMA peripheral transfer, CNT is decremented by 1. If CMEN or SBMEN in the DMA_CHxCTL register is configured to '1', CNT can be reloaded automatically to the original value at the end of transfer.</p>

12.6.7. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...7, where x is a channel number

Address offset: 0x18 + 0x18 × x

Reset value: 0x0000 0000

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

Bits	Fields	Descriptions
31:0	PADDR[31:0]	<p>Peripheral base address</p> <p>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.</p> <p>When PWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.</p> <p>Note: If PAIF in the DMA_CHxCTL register is enable, these bits must be configured to 32-bit alignment.</p>

12.6.8. Channel x memory 0 base address register (DMA_CHxM0ADDR)

x = 0...7, where x is a channel number

Address offset: 0x1C + 0x18 × x

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	M0ADDR[31:0]	<p>Memory 0 base address</p> <p>When MBS in the DMA_CHxCTL register is read as '0', these bits specific the memory base address accessed by DMA during the transmission.</p> <p>These bits can NOT be written when CHEN in the DMA_CHxCTL register is '1' and MBS in the DMA_CHxCTL register is read as '0'.</p> <p>When memory 0 is selected as memory transfer area and 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.</p> <p>When memory 0 is selected as memory transfer area and 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.</p>

12.6.9. Channel x memory 1 base address register (DMA_CHxM1ADDR)

$x = 0 \dots 7$, where x is a channel number

Address offset: $0x20 + 0x18 \times x$

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	M1ADDR[31:0]	<p>Memory 1 base address</p> <p>When MBS in the DMA_CHxCTL register is read as '1', these bits specific the memory base address accessed by DMA during the transmission.</p> <p>These bits can NOT be written when CHEN in the DMA_CHxCTL register is '1' and MBS in the DMA_CHxCTL register is read as '1'.</p> <p>When memory 1 is selected as memory tranfer area and 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.</p> <p>When memory 1 is selected as memory tranfer area and 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.</p>

12.6.10. Channel x FIFO control register (DMA_CHxFCTL)

$x = 0 \dots 7$, where x is a channel number

Address offset: $0x24 + 0x18 \times x$

Reset value: 0x0000 0000

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	FEEIE	Enable bit for FIFO error and exception interrupt

		Softw are set and clear. 0: Disable FIFO error and exception interrupt 1: Enable FIFO error and exception interrupt
6	Reserved	Must be kept at reset value.
5:3	FCNT[2:0]	FIFO counter Hardw are set and clear. 000: No data 001: One word 010: Two words 011: Three words 100: Empty 101: Full 110~111: Reserved These bits specific the number of data stored in FIFO during the transmission. When MDMEN is configured to '0', these bits has no meaning.
2	MDMEN	Multi-data mode enable Softw are set and clear. 0: Disable Multi-data mode 1: Enable Multi-data mode These bits can NOT be written when CHEN in the DMA_CHxCTL register is '1'. These bits are automatically locked as '1' by hardware immediately after enable CHEN in the DMA_CHxCTL register if TM in the DMA_CHxCTL register is configured to '10'.
1:0	FCCV[1:0]	FIFO counter critical value Softw are set and clear 00: One word 01: Two Words 10: Three Words 11: Four Words These bits can NOT be written when CHEN in the DMA_CHxCTL register is '1'. When MDMEN is configured to '0', these bits has no meaning.

12.6.11. Security status register (DMA_SSTAT)

Address offset: 0x100

Reset value: 0x0000 0000

This register may mix secure and non secure information, depending on the secure mode of each channel (SECM bit of the DMA_CHxSCTL register). A secure software can read the full interrupt status. A non-secure software is restricted to read the status of non-secure channel(s), other secure bit fields returning zero.

This register may mix privileged and unprivileged information, depending on the privileged

mode of each channel (PRIV bit of the DMA_CHxSCTL register). A privileged software can read the full interrupt status. An unprivileged software is restricted to read the status of unprivileged channel(s), other privileged bit fields returning zero.

Every status / flag bit is set by hardware, independently of the privileged and the secure mode of the channel.

Every status bit is cleared by hardware when the software sets the corresponding clear bit, in the DMA_SSC register, provided that, if the channel x is in privileged mode and/or in secure mode, then the software access to DMA_SSC is also privileged and/or secure.

31	30	29	28	27	26	25	24	23	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								IAIF7	IAIF6	IAIF5	IAIF4	IAIF3	IAIF2	IAIF1	IAIF0

r r r r r r r r

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	IAIFx	Channel x illegal access interrupt flag (x = 0...7) This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_SSCR register. 0: No illegal access event on channel x 1: An illegal access event occurred on channel x

12.6.12. Security status clear register (DMA_SSC)

Address offset: 0x104

Reset value: 0x0000 0000

This register may mix secure and non secure information, depending on the secure mode of each channel (SECM bit of the DMA_CHxSCTL register).

A secure software is able to set any flag clear bit of the DMA_SSC, and order DMA hardware to clear any corresponding flag(s) in the DMA_SSTAT register.

A non-secure software is restricted to order DMA hardware to clear the non-secure flag(s) in the DMA_SSTAT, by setting any non-secure corresponding flag clear bit(s) of the DMA_SSC register.

This register may mix privileged and unprivileged information, depending on the privileged mode of each channel (PRIV bit of the DMA_CHxSCTL register).

A privileged software is able to set any flag clear bit of the DMA_SSC, and order DMA hardware to clear any corresponding flag(s) in the DMA_SSTAT register.

An unprivileged software is restricted to order DMA hardware to clear the unprivileged flag(s) in the DMA_SSTAT, by setting any unprivileged corresponding flag clear bit(s) of the DMA_SSC register.

Writing 0 into any flag clear bit has no effect.

31	30	29	28	27	26	25	24	23	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								CIAIF7	CIAIF6	CIAIF5	CIAIF4	CIAIF3	CIAIF2	CIAIF1	CIAIF0
w								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CIAIFx	Stream x clear illegal access interrupt flag (x = 0...7) Writing 1 to this bit clears the corresponding IAIFx flag in the DMA_SSTAT register

12.6.13. Channel x security control register (DMA_CHxSCTL)

Address offset: 0x108 + 0x04 * x(x=0...7)

Reset value: 0x0000 0000

This register contains non-secure and unprivileged information: the secure state and the privileged state of the channel x (SECM and PRIV control bits). When the TZEN is cleared, all the fields in this register is force to '0x0' by hardware.

Modifying the SECM bit must be performed by a secure write access to this register (When the PRIV bit is set, a secure and privileged transfer is needed).

Modifying the PRIV bit must be performed by a privileged write access to this register (When the SECM bit is set, a secure and privileged transfer is needed).

Setting any of the DSEC or SSEC bits must be performed by a secure write access to this register when SECM is set to '1' (When the PRIV bit is set, a secure and privileged transfer is needed).

SSEC and DSEC is non-readable by a non-secure software, and non-readable by a unprivileged software if the PRIV bit is set.

31	30	29	28	27	26	25	24	23	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								PRIV	DSEC	SSEC	SECM				
w								w	w	w	w				

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	PRIV	Privileged mode This bit can only be set and cleared by a privileged software. 0: disabled

		1: enabled It must not be written when the channel is enabled (CHEN = 1). It is read-only when the channel is enabled (CHEN = 1).
2	DSEC	<p>Security of the DMA transfer to the destination</p> <p>This bit can only be read, set or cleared by a secure software. It must be a privileged software if the channel is in privileged mode.</p> <p>This bit is cleared by hardware when the securely written data bit 0 is cleared (on a secure reconfiguration of the channel as non-secure).</p> <p>A non-secure read to this secure configuration bit returns 0.</p> <p>A non-secure write of 1 to this secure configuration bit has no impact on the register setting and an illegal access pulse is asserted.</p> <p>Destination (peripheral or memory) of the DMA transfer is defined by the direction DIR configuration bit.</p> <p>0: non-secure DMA transfer to the destination</p> <p>1: secure DMA transfer to the destination</p> <p>This bit must not be written when the channel is enabled (CHEN = 1). It is read-only when the channel is enabled (CHEN = 1).</p>
1	SSEC	<p>Security of the DMA transfer from the source</p> <p>This bit can only be accessed – read, set or cleared – by a secure software. It must be a privileged software if the channel is in privileged mode.</p> <p>This bit is cleared by hardware when the securely written data bit 0 is cleared (on a secure reconfiguration of the channel as non-secure).</p> <p>A non-secure read to this secure configuration bit returns 0.</p> <p>A non-secure write of 1 to this secure configuration bit has no impact on the register setting and an illegal access pulse is asserted.</p> <p>Source (peripheral or memory) of the DMA transfer is defined by the direction DIR configuration bit.</p> <p>0: non-secure DMA transfer from the source</p> <p>1: secure DMA transfer from the source</p> <p>This bit must not be written when the channel is enabled (CHEN = 1). It is read-only when the channel is enabled (CHEN = 1).</p>
0	SECM	<p>Secure mode</p> <p>This bit can only be set or cleared by a secure software.</p> <p>0: non-secure channel</p> <p>1: secure channel</p> <p>This bit must not be written when the channel is enabled (CHEN = 1). It is read-only when the channel is enabled (CHEN = 1).</p>

13. Debug (DBG)

13.1. Overview

The GD32W51x 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) and trace functions in addition to standard JTAG debug. The debug and trace functions refer to the following documents:

- Cortex-M33 Technical Reference Manual
- ARM Debug Interface v5 Architecture Specification

The DBG hold unit helps debugger to debug power saving mode, TIMER, I2C, WWDGT, FWDGT and RTC. When corresponding bit is set, provide clock when in power saving mode or hold the state for TIMER, WWDGT, FWDGT, I2C or RTC.

13.2. JTAG/SW function description

Debug capabilities can be accessed by a debug tool via Serial Wire (SW - Debug Port) or JTAG interface (JTAG - Debug Port).

13.2.1. Switch JTAG or SW interface

By default, the JTAG interface is active. The sequence for switching from JTAG to SWD is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 11100111001110 (0xE79E LSB first).
- Send 50 or more TCK cycles with TMS = 1.

The sequence for switching from SWD to JTAG is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 111001110011100 (0xE73C LSB first).
- Send 50 or more TCK cycles with TMS = 1.

13.2.2. Pin assignment

The JTAG interface provides 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) provide 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK). The two SW pin are multiplexed with two of five JTAG pin, which is SWDIO multiplexed with JTMS, SWCLK multiplexed with JTCK. The JTDO is also used as Trace async data output (TRACESWO) when async trace enabled.

The pin assignment are:

PA15	:	JTDI
PA14	:	JTCK/SWCLK
PA13	:	JTMS/SWDIO
PB4	:	NJTRST
PB3	:	JTDO

By default, 5-pin standard JTAG debug mode is chosen after reset. Users can also use JTAG function without NJTRST pin, then the PB4 can be used to other GPIO functions. (NJTRST tied to 1 by hardware). If switch to SW debug mode, the PA15/PB4/PB3 are released to other GPIO functions. If JTAG and SW not used, all 5-pin can be released to other GPIO functions. Please refer to [GPIO pin configuration](#).

13.2.3. JTAG daisy chained structure

The Cortex-M33 JTAG TAP is connected to a Boundary-Scan (BSD) JTAG TAP. The BSD JTAG IR is 5-bit width, while the Cortex-M33 JTAG IR is 4-bit width. So when JTAG in IR shift step, it first shift 5-bit BYPASS instruction (5'b 11111) for BSD JTAG, and then shift normal 4-bit instruction for Cortex-M33 JTAG. Because of the data shift under BSD JTAG BYPASS mode, adding 1 extra bit to the data chain is needed.

The BSD JTAG IDCODE is 0x790007A3.

13.2.4. Debug reset

The JTAG-DP and SW-DP register 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, it can perform debug feature under system reset. Such as, halt-after-reset, which is the debugger sets halt under system reset, and the core halts immediately after the system reset is released.

13.2.5. JEDEC-106 ID code

The Cortex-M33 integrates JEDEC-106 ID code, which is located in ROM table and mapped on the address of 0xE00FF000_0xE00FFFFF.

13.3. Debug hold function description

13.3.1. Debug support for power saving mode

When STB_HOLD bit in DBG control register 0 (DBG_CTL0) is set and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC16M, and the debugger can debug in standby mode. When exit the standby mode, a system reset generated.

When DSLP_HOLD bit in DBG control register 0 (DBG_CTL0) is set and entering the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC16M, and the debugger can debug in Deep-sleep mode.

When SLP_HOLD bit in DBG control register 0 (DBG_CTL0) 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.

13.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT and RTC

When the core halted and the corresponding bit in DBG control register 1/2 (DBG_CTL1/2) is set, the following behaved.

For TIMER, the timer counters stopped and hold for debug.

For I2C, SMBUS timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For RTC, the counter is stopped for debugging.

13.4. DBG registers

DEBUG base address: 0xE0044000

13.4.1. ID code register (DBG_ID)

Address offset: 0x00

Reset value: 0x0000 0000; Read only

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



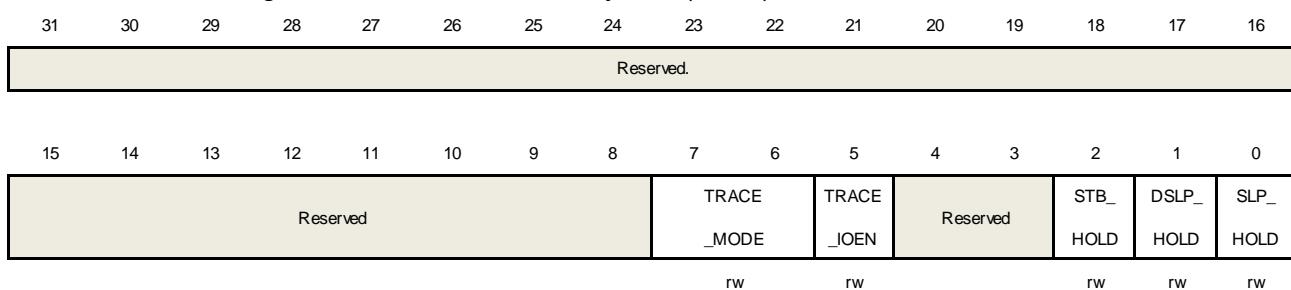
Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits read by software, These bits are unchanged constant

13.4.2. Control register 0 (DBG_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000; power reset only

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	TRACE_MODE[1:0]	Trace pin allocation mode This bit is set and reset by software 00: Trace pin used in asynchronous mode 01: Trace pin used in synchronous mode and the data length is 1 10: Trace pin used in synchronous mode and the data length is 2

11: Trace pin used in synchronous mode and the data length is 4

5	TRACE_IOEN	Trace pin allocation enable This bit is set and reset by softw are 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 register This bit is set and reset by softw are 0: no effect 1: At the standby mode, the clock of AHB bus and system clock are provided by CK_IRC16M, a system reset generated when exit standby mode
1	DSLP_HOLD	Deep-sleep mode hold register This bit is set and reset by softw are 0: no effect 1: At the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC16M
0	SLP_HOLD	Sleep mode hold register This bit is set and reset by softw are 0: no effect 1: At the sleep mode, the clock of AHB is on.

13.4.3. Control register 1 (DBG_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000; power reset only

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.										I2C1_HO LD	I2C0_HO LD	Reserved.				
rw	rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved.	FWDGT_ HOLD	WWWDGT_ _HOLD	RTC_HO LD	Reserved					TIMER5_ HOLD	TIMER4_ HOLD	TIMER3_ HOLD	TIMER2_ HOLD	TIMER1_ HOLD			
rw	rw	rw	rw						rw	rw	rw	rw	rw			

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	I2C1_HOLD	I2C1 hold bit This bit is set and reset by softw are

		0: no effect 1: hold the I2C1 SMBUS timeout for debug when core halted
21	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software 0: no effect 1: hold the I2C0 SMBUS timeout for debug when core halted
20:13	Reserved	Must be kept at reset value
12	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.
11	WWDGTHOLD	WWDGTHOLD 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.
10	RTC_HOLD	RTC hold bit This bit is set and reset by software. 0: no effect 1: hold the RTC counter for debugging when the core is halted.
9:5	Reserved	Must be kept at reset value
4	TIMER5_HOLD	TIMER5 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER5 counter for debugging when the core is halted.
3	TIMER4_HOLD	TIMER4 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER4 counter for debugging when the core is halted.
2	TIMER3_HOLD	TIMER3 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER3 counter for debugging when the core is halted.
1	TIMER2_HOLD	TIMER2 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER2 counter for debugging when the core is halted.
0	TIMER1_HOLD	TIMER1 hold bit This bit is set and reset by software.

0: no effect

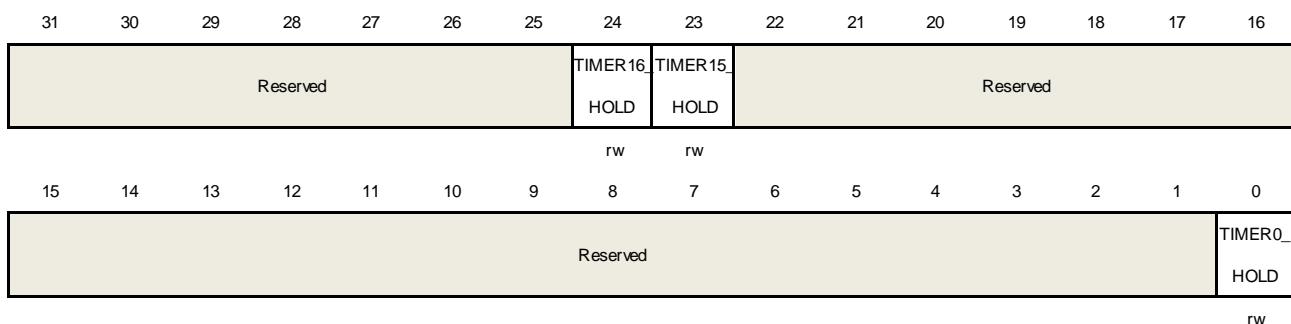
1: hold the TIMER1 counter for debugging when the core is halted.

13.4.4. Control register 2 (DBG_CTL2)

Address offset: 0x0C

Reset value: 0x0000 0000; power reset only

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



Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	TIMER16_HOLD	<p>TIMER16 hold bit</p> <p>This bit is set and reset by software.</p> <p>0: no effect</p> <p>1: hold the TIMER16 counter for debugging when the core is halted.</p>
23	TIMER15_HOLD	<p>TIMER15 hold bit</p> <p>This bit is set and reset by software.</p> <p>0: no effect</p> <p>1: hold the TIMER15 counter for debugging when the core is halted.</p>
22:1	Reserved	Must be kept at reset value.
0	TIMER0_HOLD	<p>TIMER0 hold bit</p> <p>This bit is set and reset by software</p> <p>0: no effect</p> <p>1: hold the TIMER0 counter for debug when core halted</p>

14. Analog to digital converter (ADC)

14.1. Overview

The 12-bit ADC is an analog-to-digital converter using the successive approximation method. The ADC includes 9 external channels, 2 internal channels and the battery voltage (V_{BAT}) channel that can convert analog signals. Analog watchdog allows the application to detect whether the input voltage exceeds the user's set of high and low threshold. The A/D conversion of each channel can be performed in single, continuous, scan, or discontinuous mode. A left-aligned or right-aligned 16-bit data register holds the output of the ADC. An on-chip hardware oversample scheme improves performances while off-loading the related computational burden from the MCU.

14.2. Characteristics

- High performance:
 - 2.5MSPs for 12-bit resolution
 - Programmable sampling time
 - Data alignment with built-in data registers
 - DMA support for regular channels and inserted channels
- Analog input channels:
 - 9 external analog inputs
 - 1 channel for internal temperature sensor (V_{SENSE})
 - 1 channel for internal reference voltage (V_{REFINT})
 - 1 channel for external battery power supply pin (V_{BAT})
- Start-of-conversion can be initiated:
 - By software
 - By hardware triggers
- Conversion modes:
 - Convert a single channel or scan a sequence of channels
 - Single mode converts selected input channel once per trigger
 - Continuous mode converts selected input channels continuously
 - Discontinuous mode
- Interrupt generation
 - At the end of regular or inserted group conversions
 - Analog watchdog event
 - Overflow event
- Analog watchdog
- Oversampling
 - 16-bit data register

- Oversampling ratio adjustable from 2x to 256x
- Programmable data shift up to 8-bits
- ADC supply requirements: 2.5V to 3.6V, and typical power supply voltage is 3.3V.
- ADC input range: $0 \leq V_{IN} \leq V_{DDA}$

14.3. Pins and internal signals

[**Figure 14-1. ADC module block diagram**](#) shows the ADC block diagram. [**Table 14-1. ADC internal signals**](#) and [**Table 14-2. ADC pins definition**](#) give the ADC internal signals and pins description.

Table 14-1. ADC internal signals

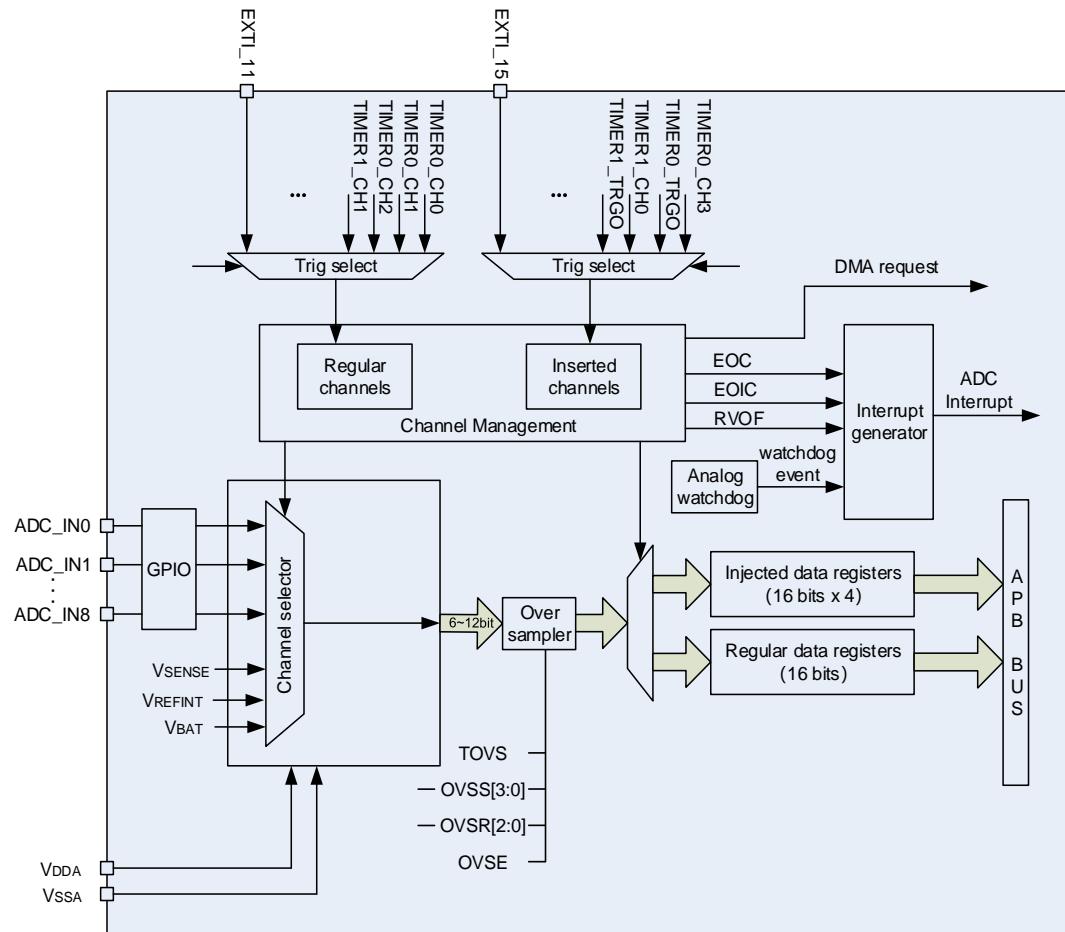
Internal signal name	Signal type	Description
V_{SENSE}	Input	Internal temperature sensor output voltage
V_{REFINT}	Input	Internal voltage reference output voltage

Table 14-2. ADC pins definition

Name	Signal type	Remarks
V_{DDA}	Input, analog power supply	Analog power supply equals to V_{DD} , and $2.5V \leq V_{DDA} \leq 3.6V$
V_{SSA}	Input, analog power supply ground	Ground for analog power supply equals to V_{SS}
$ADCx_IN[8:0]$	Input, Analog signals	Up to 9 external channels
V_{BAT}	Input, Analog signals	External battery voltage

14.4. Function overview

Figure 14-1. ADC module block diagram



14.4.1. ADC clock

The ADCCLK clock provided by the clock controller is synchronous with the AHB and APB2 clock. The maximum frequency is 35MHz. The RCU controller has a dedicated programmable prescaler for the ADC clock.

14.4.2. ADCON switch

The ADC module is enabled or disabled by configuring the ADCON bit in the ADC_CTL1 register. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is 0, the analog sub-module will be enter power-down mode

14.4.3. Regular and inserted channel groups

The ADC supports 12 multiplexed channels and organizes the conversion results into two groups: a regular channel group and an inserted channel group.

In the regular group, a sequence of up to 9 conversions can be organized in a specific sequence. The ADC_RSQ0~ADC_RSQ2 registers specify the selected channels of the regular group. The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length.

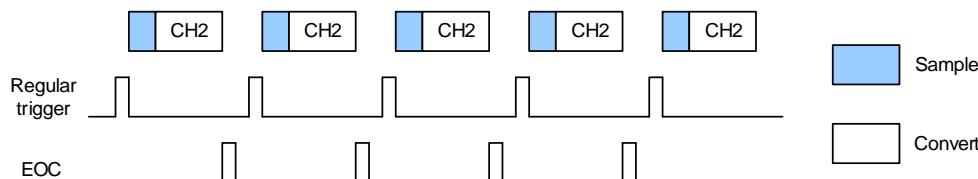
In the inserted group, a sequence of up to 4 conversions can be organized in a specific sequence. The ADC_ISQ register specifies the selected channels of the inserted group. The IL[1:0] bits in the ADC_ISQ register specify the total conversion sequence length.

14.4.4. Conversion modes

Single conversion mode

This mode can be used in both regular and inserted channel groups. In the single conversion mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits in ADC_RSQ2 or the channel specified in the ISQ3[4:0] bits in ADC_ISQ. When the ADCON is 1, the ADC samples and converts a single channel, once the corresponding software trigger or external trigger is active.

Figure 14-2. Single conversion mode



After conversion of a single regular channel, the conversion data will be stored in the ADC_RDATA register, the EOC will be set. An interrupt will be generated if the EOCIE bit is set.

After conversion of a single injected channel, the conversion data will be stored in the ADC_IDATA0 register, the EOC and EOIC will be set. An interrupt will be generated if the EOCIE or EOICIE bit is set.

Software procedure for a single conversion of a regular channel:

1. Make sure the DISRC, SM in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset;
2. Configure the RSQ0 with the analog channel number;
3. Configure the ADC_SAMPTx register;
4. Configure the ETMRC and ETSRC bits in the ADC_CTL1 register if it is needed;
5. Set the SWRCST bit, or generate an external trigger for the regular group;
6. Wait for the EOC flag to be set;
7. Read the converted result from the ADC_RDATA register;
8. Clear the EOC flag by writing 0.

Software procedure for a single conversion of an inserted channel:

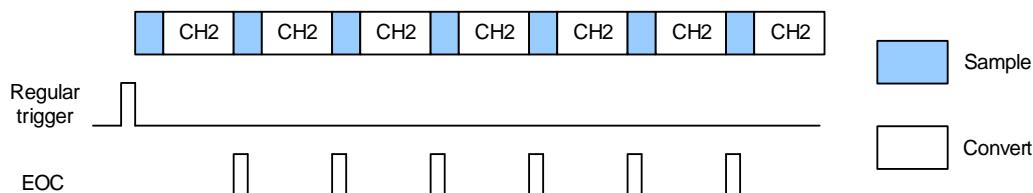
1. Make sure the DISIC, SM in the ADC_CTL0 register are reset;

2. Configure ISQ3 with the analog channel number;
3. Configure ADC_SAMPTx register;
4. Configure the ETMIC and ETSIC bits in the ADC_CTL1 register if it is needed;
5. Set the SWICST bit, or generate an external trigger for the inserted group;
6. Wait for the EOC/ EOIC flags to be set;
7. Read the converted result in the ADC_IDATA0 register;
8. Clear the EOC/ EOIC flags by writing 0.

Continuous conversion mode

This mode can be used in the regular channel group. The continuous conversion mode will be enabled when the CTN bit in the ADC_CTL1 register is set. In this mode, the ADC performs conversion on the channel specified in the RSQ0[4:0]. When the ADCON is 1, the ADC samples and converts specified a channel, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA register.

Figure 14-3. Continuous conversion mode



Software procedure for continuous conversion on a regular channel:

1. Set the CTN bit in the ADC_CTL1 register;
2. Configure RSQ0 with the analog channel number;
3. Configure ADC_SAMPTx register;
4. Configure ETMRC and ETSRC bits in the ADC_CTL1 register if it is needed;
5. Set the SWRCST bit, or generate an external trigger for the regular group;
6. Wait for the EOC flag to be set;
7. Read the converted result in the ADC_RDATA register;
8. Clear the EOC flag by writing 0;
9. Repeat steps 6~8 as soon as the conversion is in need.

To avoid checking, DMA can be used to transfer the converted data:

1. Set the CTN and DMA bits in the ADC_CTL1 register;
2. Configure RSQ0 with the analog channel number;
3. Configure ADC_SAMPTx register;
4. Configure the ETMRC and ETSRC bits in the ADC_CTL1 register in if it is needed;
5. Prepare the DMA module to transfer data from the ADC_RDATA;
6. Set the SWRCST bit, or generate an external trigger for the regular group.

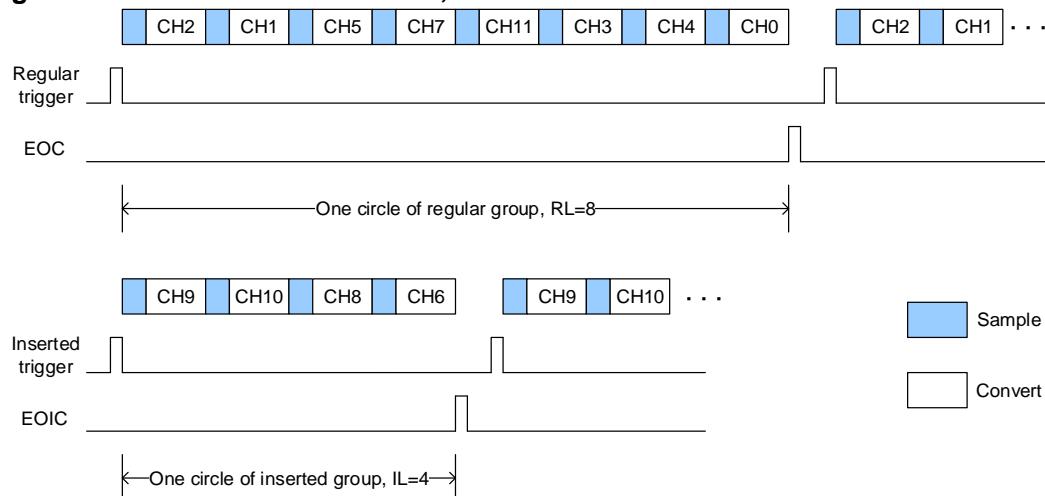
Scan conversion mode

The scan conversion mode will be enabled when the SM bit in the ADC_CTL0 register is set. In this mode, the ADC performs conversion on the channels with a specific sequence

specified in the ADC_RSQ0~ADC_RSQ2 registers or ADC_ISQ register. When the ADCON is 1, the ADC samples and converts specified channels one by one in the regular or inserted group till the end of the regular or inserted group, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA or ADC_IDATAx register. After conversion of the regular or inserted channel group, the EOC or EOIC will be set. An interrupt will be generated if the EOCIE or EOICIE bit is set. The DMA bit in ADC_CTL1 register must be set when the regular channel group works in scan mode.

After conversion of a regular channel group, the conversion can be restarted automatically if the CTN bit in the ADC_CTL1 register is set.

Figure 14-4. Scan conversion mode, continuous disable

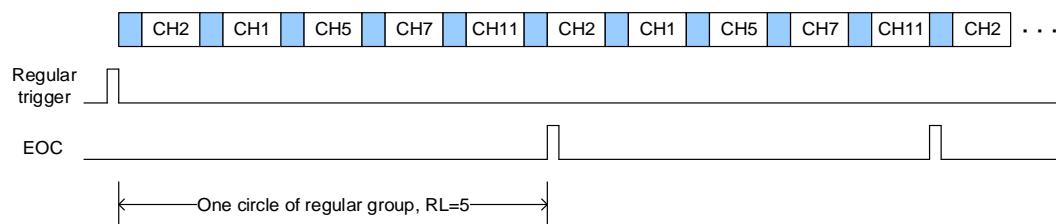


Software procedure for scan conversion on a regular channel group:

1. Set the SM bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register;
2. Configure ADC_RSQx and ADC_SAMPTx registers;
3. Configure the ETMRC and ETSRC bits in the ADC_CTL1 register if it is needed;
4. Prepare the DMA module to transfer data from the ADC_RDATA;
5. Set the SWRCST bit, or generate an external trigger for the regular group;
6. Wait for the EOC flag to be set;
7. Clear the EOC flag by writing 0.

Software procedure for scan conversion on an inserted channel group:

1. Set the SM bit in the ADC_CTL0 register;
2. Configure ADC_ISQ and ADC_SAMPTx registers;
3. Configure ETMIC and ETSIC bits in the ADC_CTL1 register if it is needed;
4. Set the SWICST bit, or generate an external trigger for the inserted group;
5. Wait the EOC/EOIC flags to be set;
6. Read the converted result in the ADC_IDATAx register;
7. Clear the EOC/EOIC flag by writing 0 to them.

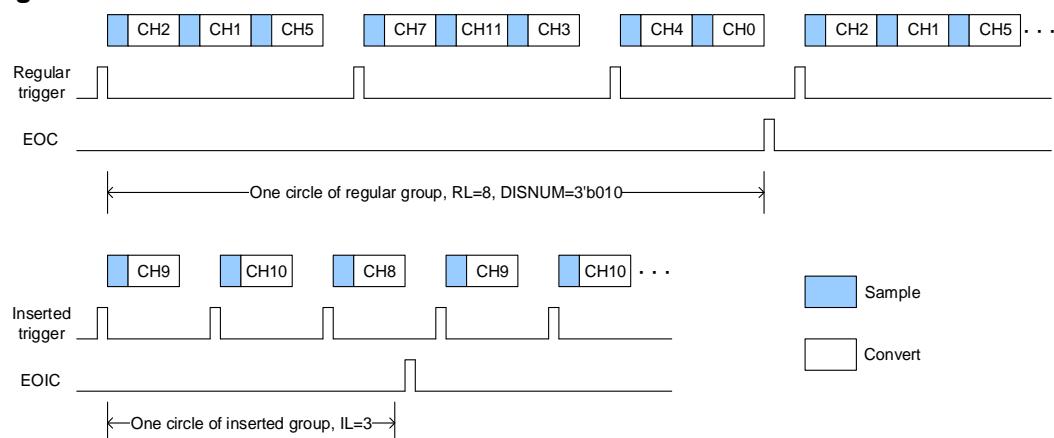
Figure 14-5. Scan conversion mode, continuous enable


Discontinuous mode

For regular channel group, the discontinuous conversion mode will be enabled when the DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions ($n \leq 8$) which is a part of the sequence of conversions selected in the ADC_RSQ0~ADC_RSQ2 registers. The value of n is defined by the DISNUM[2:0] bits in the ADC_CTL0 register. When the corresponding software trigger or external trigger is active, the ADC samples and converts the next n channels selected in the ADC_RSQ0~ADC_RSQ2 registers until all the channels in the regular sequence are done. The EOC will be set after every circle of the regular channel group. An interrupt will be generated if the EOICIE bit is set.

For inserted channel group, the discontinuous conversion mode will be enabled when the DISIC bit in the ADC_CTL0 register is set. In this mode, the ADC performs one conversion which is a part of the sequence of conversions selected in the ADC_ISQ register. When the corresponding software trigger or external trigger is active, the ADC samples and converts the next channel selected in the ADC_ISQ register until all the channels in the inserted sequence are done. The EOIC will be set after every circle of the inserted channel group. An interrupt will be generated if the EOICIE bit is set.

The regular and inserted groups cannot both work in discontinuous conversion mode. Only one group conversion can be set in discontinuous conversion mode at a time.

Figure 14-6. Discontinuous conversion mode


Software procedure for discontinuous conversion on a regular channel group:

1. Set the DISRC bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register;

2. Configure DISNUM [2:0] bits in the ADC_CTL0 register;
3. Configure ADC_RSQx and ADC_SAMPTx registers;
4. Configure the ETMRC and ETSRC bits in the ADC_CTL1 register if it is needed;
5. Prepare the DMA module to transfer data from the ADC_RDATA;
6. Set the SWRCST bit, or generate an external trigger for the regular group;
7. Repeat step6 if it is needed;
8. Wait the EOC flag to be set;
9. Clear the EOC flag by writing 0.

Software procedure for discontinuous conversion on an inserted channel group:

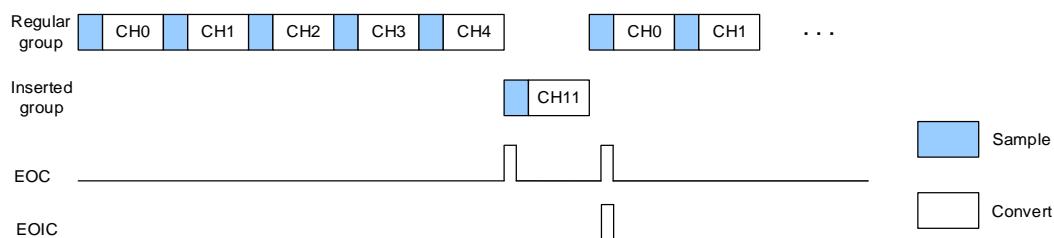
1. Set the DISIC bit in the ADC_CTL0 register;
2. Configure ADC_ISQ and ADC_SAMPTx registers;
3. Configure ETMIC and ETSIC bits in the ADC_CTL1 register if it is needed;
4. Set the SWICST bit, or generate an external trigger for the inserted group;
5. Repeat step4 if it is needed;
6. Wait the EOC/EOIC flags to be set;
7. Read the converted result in the ADC_IDATAx register;
8. Clear the EOC/EOIC flag by writing 0.

14.4.5. Inserted channel management

Auto-insertion

The inserted group channels are automatically converted after the regular group channels when the ICA bit in ADC_CTL0 register is set. In this mode, the external trigger on inserted channels cannot be enabled. A sequence of up to 13 conversions programmed in the ADC_RSQ0~ADC_RSQ2 and ADC_ISQ registers can be used to convert in this mode. In addition to the ICA bit, if the CTN bit is also set, regular channels are continuously converted after inserted channels.

Figure 14-7. Auto-insertion, CTN = 1



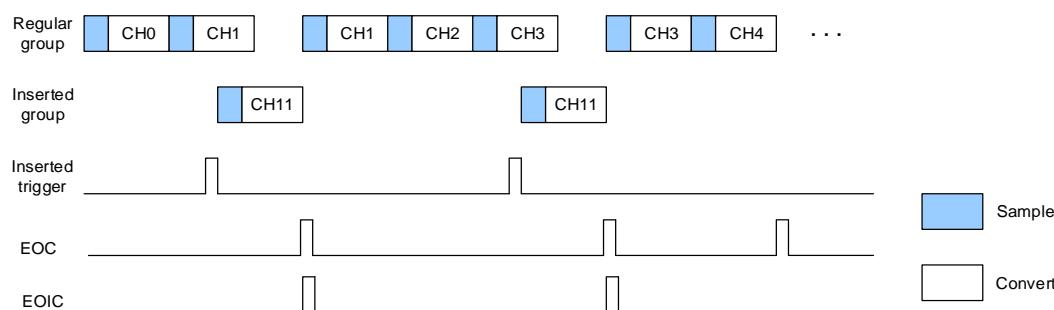
The auto insertion mode can not be enabled when the discontinuous conversion mode is set.

Triggered insertion

If the ICA bit is cleared, the triggered insertion occurs if a software or external trigger occurs during the regular channel group conversion. In this situation, the ADC aborts the current conversion and starts the conversion of inserted channelgroup. After the inserted channel group is done, the regular channel group conversion will resume from the last aborted

conversion.

Figure 14-8. Triggered insertion



14.4.6. Analog watchdog

The analog watchdog is enabled when the RWDEN and IWDEN bits in the ADC_CTL0 register are set for regular and inserted channel groups respectively. When the analog voltage converted by the ADC is below the low threshold or above the high threshold, the WDE bit in ADC_STAT register will be set. An interrupt will be generated if the WDEIE bit is set. The ADC_WDHT and ADC_WDLT registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold value is independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are selected by the RWDEN, IWDEN, WDSC and WDCHSEL [4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog.

14.4.7. Data alignment

The alignment of data stored after conversion can be specified by DAL bit in the ADC_CTL1 register.

After being decreased by the user-defined offset written in the ADC_IOFFx registers, the inserted group data value may be a negative value. The sign value is extended.

When left-aligned, the 12 data are aligned on a half-word basis as shown below [Figure 14-9. Data alignment of 12-bit resolution](#).

Figure 14-9. Data alignment of 12-bit resolution

Regular group data															
0 0 0 0 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0															
Inserted group data															
Sign Sign Sign Sign D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0															
DAL=0															
Regular group data															
D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0 0															
Inserted group data															
Sign D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0															
DAL=1															

14.4.8. Programmable sampling time

The number of ADC_CLK cycles which is used to sample the input voltage can be specified by the SPTn[2:0] bits in the ADC_SAMPT0 and ADC_SAMPT1 registers. And each channel can specify different sampling times. For 12-bit resolution, the total conversion time is “sampling time + 12.5” ADCCLK cycles.

Example:

ADCCLK = 35MHz and sample time is 1.5 cycles, the total conversion time is “1.5+12.5” ADCCLK cycles, that means 0.4us.

14.4.9. External trigger

The conversion of regular or inserted group can be triggered by rising/ falling edge of external trigger inputs. The ETMRC[1:0] and ETMIC[1:0] bits in the ADC_CTL1 register control the trigger modes of regular and inserted group respectively. The external trigger source of regular channel group is controlled by the ETSRC[3:0] bits in the ADC_CTL1 register, while the external trigger source of inserted channel group is controlled by the ETSIC[3:0] bits in the ADC_CTL1 register

The ETSRC[3:0] and ETSIC[3:0]control bits are used to specify which one can trigger conversion for the regular and inserted groups.

Table 14-3. External trigger modes

ETMRC[1:0]/ ETMIC[1:0]	Trigger mode
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

Table 14-4. External trigger for regular channels of ADC

ETSRC[3:0]	Trigger source	Trigger type
0000	TIMER0_CH0	Internal on-chip signal
0001	TIMER0_CH1	
0010	TIMER0_CH2	
0011	TIMER1_CH1	
0100	TIMER1_CH2	
0101	TIMER1_CH3	
0110	TIMER1_TRGO	
0111	TIMER2_CH0	
1000	TIMER2_TRGO	
1001	TIMER3_CH3	
1010	TIMER4_CH0	
1011	TIMER4_CH1	
1100	TIMER4_CH2	
1101	Reserved	
1110	Reserved	
1111	EXTI_11	External signal

Table 14-5. External trigger for inserted channels of ADC

ETSIC[3:0]	Trigger source	Trigger type
0000	TIMER0_CH3	Internal on-chip signal
0001	TIMER0_TRGO	
0010	TIMER1_CH0	
0011	TIMER1_TRGO	
0100	TIMER2_CH1	
0101	TIMER2_CH3	
0110	TIMER3_CH0	
0111	TIMER3_CH1	
1000	TIMER3_CH2	
1001	TIMER3_TRGO	
1010	TIMER4_CH3	
1011	TIMER4_TRGO	
1100	Reserved	
1101	Reserved	
1110	Reserved	
0110	EXTI_15	External signal

The selection of the external triggers can be changed on the fly, while no trigger event occurs due to this change.

14.4.10. DMA request

The DMA request is used to transfer data for conversion of more than one channel. The DMA request of regular channel is enabled by the DMA bit of ADC_CTL1 register. When this bit is set, the ADC generates a DMA request at the end of conversion of a regular channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA register to the destination location which is specified by the user.

14.4.11. Overflow detection

Overflow detection is enabled when DMA is enabled or EOQM bit in ADC_CTL1 is set. An overflow event occurs when a regular conversion is done before the prior regular data has been read out. The ROVF bit of the ADC_STAT is set. Overflow interrupt is generated if the ROVFIIE bit in the ADC_CTL0 is set.

It is recommended to reinitialize the DMA module to recover the ADC from ROVF state. To ensure the regular converted data are transferred correctly, the internal state machine is reset. The ADC conversion will be stalled until the ROVF bit is cleared.

Software procedure for recovering the ADC from ROVF state:

1. Clear DMA bit of ADC_CTL1 to 0
2. Clear ADCON bit of ADC_CTL1 to 0
3. Clear CHEN bit of DMA_CHxCTL to 0 with reinit DMA module
4. Clear ROVF bit of ADC_STAT to 0
5. Set CHEN bit of DMA_CHxCTL to 1
6. Set DMA bit of ADC_CTL1 to 1
7. Set ADCON bit of ADC_CTL1 to 1
8. Wait T(setup)
9. Start conversion with software or trigger.

14.4.12. Temperature sensor, internal reference voltage VREFINT and external battery voltage VBAT

When the TSVREN bit in ADC_CCTL register is set, the temperature sensor channel (ADC_IN9) and V_{REFINT} channel (ADC_IN10) are enabled. 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 (refer to device datasheet for more information). When this sensor is not in use, it can be set in power down mode by resetting the TSVREN bit.

The output voltage of the temperature sensor changes linearly with temperature. Because there is an offset, which is up to 45°C and varies from chip to chip due to process variation, the internal temperature sensor is more suitable for applications that detect temperature

variations than absolute temperature. When it is used to detect accurate temperature, an external temperature sensor part should be used to calibrate the offset error.

The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and comparators. V_{REFINT} is internally connected to the ADC_IN10 input channel.

When the VBATEN bit of ADC_CCTL register is set, the external battery voltage can be detected by ADC_IN11. To ensure the the V_{BAT} voltage is no higher than V_{DDA} , the battery voltage is internal divided by 4.

To use the temperature sensor:

1. Configure the conversion sequence (ADC_IN10) and the sampling time (t_{s_temp} μ s) for the channel.
2. Enable the temperature sensor by setting the TSVREN bit in ADC_CCTL.
3. Start the ADC conversion by setting the ADCON bit (or by external trigger).
4. Read the resulting temperature data ($V_{temperature}$) in the ADC data register, and get the temperature using the following formula:

$$\text{Temperature } (^{\circ}\text{C}) = \{(V_{25} - V_{temperature} \text{ (digit)}) / \text{Avg_Slope}\} + 25.$$

V_{25} : $V_{temperature}$ value at 25°C, please refer to device datasheet for more information.

Avg_Slope: Average Slope for curve between Temperature vs. $V_{temperature}$, the typical value please refer to device datasheet.

14.4.13. On-chip hardware oversampling

The on-chip hardware oversampling unit, which is enabled by OVSEN bit in the ADC_OVSAMPCTL register, provides higher data resolution at the cost of lower output data rate.

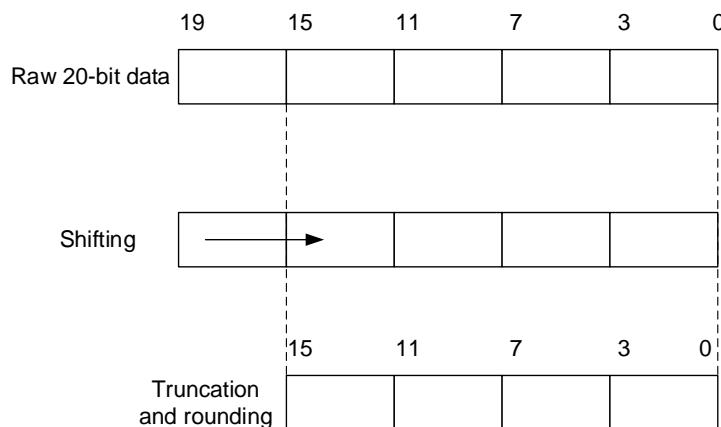
It provides a result with the following form, where N and M can be adjusted, and $D_{OUT}(n)$ is the n-th output digital signal of the ADC:

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{OUT}(n) \quad (14-1)$$

The on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting. The oversampling ratio N is defined using the OVSR[2:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to 256x. The division coefficient M consists of a right bit shift up to 8 bits. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

The summation unit can yield a result up to 20 bits (256 x 12-bit), which is first shifted right. The upper bits of the result are then truncated, keeping only the 16 least significant bits rounded to the nearest value using the least significant bits left apart by the shifting, before being finally transferred into the data register.

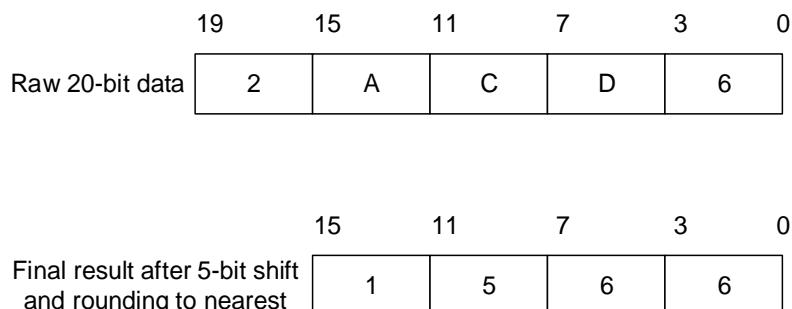
Figure 14-10. 20-bit to 16-bit result truncation



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

[**Figure 14-11. A numerical example with 5-bit shifting and rounding**](#) shows a numerical example of the processing, from a raw 20-bit accumulated data to the final 16-bit result.

Figure 14-11. A numerical example with 5-bit shifting and rounding



[**Table 14-6. Maximum output results for N and M combinations \(grayed values indicates truncation\)**](#) below gives the data format for the various N and M combinations, and the raw conversion data equals 0xFFFF.

Table 14-6. Maximum output results for N and M combinations (grayed values indicates truncation)

Oversampling ratio	Max Raw data	No-shift OVSS= 0000	1-bit shift OVSS= 0001	2-bit shift OVSS= 0010	3-bit shift OVSS= 0011	4-bit shift OVSS= 0100	5-bit shift OVSS= 0101	6-bit shift OVSS= 0110	7-bit shift OVSS= 0111	8-bit shift OVSS= 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFFF0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF

Oversampling ratio	Max Raw data	No-shift OVSS= 0000	1-bit shift OVSS= 0001	2-bit shift OVSS= 0010	3-bit shift OVSS= 0011	4-bit shift OVSS= 0100	5-bit shift OVSS= 0101	6-bit shift OVSS= 0110	7-bit shift OVSS= 0111	8-bit shift OVSS= 1000
32x	0x1FFE0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

When compared to standard conversion mode, the conversion timings of oversampling mode do not change, and the sampling time is maintained the same as that of standard conversion mode during the whole oversampling sequence. New data are provided every N conversion, with an equivalent delay equal to:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (14-2)$$

14.4.14. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for regular and inserted groups
- The analog watchdog event (the analog watchdog status bit is set)
- Overflow event

Separate interrupt enable bits are available for flexibility. The interrupts of ADC is mapped into the interrupt vector ISR[18].

14.5. Register definition

ADC secure access base address: 0x5001 2000
 ADC non-secure access base address: 0x4001 2000

14.5.1. Status register (ADC_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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ROVF	STRC	STIC	EOIC	EOC	WDE		
rc_w0								rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	ROVF	Regular data register overflow 0: Regular data register not overflow 1: Regular data register overflow This bit is set by hardware when the regular data registers are overflow, in single mode or multi mode. This flag is only set when DMA is enabled or end of conversion mode is set to 1(EOCM=1). The recent regular data is lost when this bit is set. Cleared by software writing 0 to it.
4	STRC	Start flag of regular channel group 0: No regular channel group started 1: Regular channel group started Set by hardware when regular channel conversion starts. Cleared by software writing 0 to it.
3	STIC	Start flag of inserted channel group 0: No inserted channel group started 1: Inserted channel group started Set by hardware when inserted channel group conversion starts. Cleared by software writing 0 to it.
2	EOIC	End of inserted group conversion flag 0: No end of inserted group conversion 1: End of inserted group conversion

Set by hardw are at the end of all inserted group channel conversion.

Cleared by softw are writing 0 to it.

1	EOC	End of group conversion flag 0: No end of group conversion 1: End of group conversion Set by hardw are at the end of a regular or inserted group channel conversion. Cleared by softw are writing 0 to it or by reading the ADC_RDATA register.
0	WDE	Analog watchdog event flag 0: No analog watchdog event 1: Analog watchdog event Set by hardw are when the converted voltage crosses the values programmed in the ADC_WDLT and ADC_WDHT registers. Cleared by softw are writing 0 to it.

14.5.2. 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
Reserved			ROVFIE	Reserved		RWDEN	IWDEN	Reserved			Reserved				
rw				rw		rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]			DISIC	DISRC	ICA	WDSC	SM	EOICIE	WDEIE	EOCIE	WDCHSEL [4:0]				
rw			rw	rw	rw	rw	rw	rw	rw	rw					rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	ROVFIE	Interrupt enable for ROVF 0: ROVF interrupt disable 1: ROVF interrupt enable
25:24	Reserved	Must be kept at reset value.
23	RWDEN	Regular channel analog watchdog enable 0: Regular channel analog watchdog disable 1: Regular channel analog watchdog enable
22	IWDEN	Inserted channel analog watchdog enable 0: Inserted channel analog watchdog disable 1: Inserted channel analog watchdog enable

21:16	Reserved	Must be kept at reset value.
15:13	DISNUM[2:0]	Number of conversions in discontinuous mode The number of channels to be converted after a trigger will be DISNUM [2:0] +1
12	DISIC	Discontinuous mode on inserted channels 0: Discontinuous mode on inserted channels disable 1: Discontinuous mode on inserted channels enable
11	DISRC	Discontinuous mode on regular channels 0: Discontinuous mode on regular channels disable 1: Discontinuous mode on regular channels enable
10	ICA	Inserted channel group convert automatically 0: Inserted channel group convert automatically disable 1: Inserted channel group convert automatically enable
9	WDSC	When in scan mode, analog watchdog is effective on a single channel 0: Analog watchdog is effective on all channels 1: Analog watchdog is effective on a single channel
8	SM	Scan mode 0: Scan mode disable 1: Scan mode enable
7	EOCIE	Interrupt enable for EOC 0: EOC interrupt disable 1: EOC interrupt enable
6	WDEIE	Interrupt enable for WDE 0: WDE interrupt disable 1: WDE interrupt enable
5	EOCIE	Interrupt enable for EOC 0: EOC interrupt disable 1: EOC interrupt enable
4:0	WDCHSEL[4:0]	Analog watchdog channel select 00000: ADC channel 0 00001: ADC channel 1 00010: ADC channel 2 00011: ADC channel 3 00100: ADC channel 4 00101: ADC channel 5 00110: ADC channel 6 00111: ADC channel 7 01000: ADC channel 8 01001: ADC channel 9 01010: ADC channel 10

01011: ADC channel 11

Other values are reserved.

Note: ADC analog inputs Channel 9, Channel 10 and Channel 11 are internally connected to the temperature sensor and V_{REFINT} and V_{BAT} analog inputs.

14.5.3. Control register 1 (ADC_CTL1)

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	SWRCST	ETMRC[1:0]		ETSRC[3:0]	Reserved	SWICST	ETMIC[1:0]		ETSIC[3:0]						
	rw	rw		rw		rw	rw		rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	DAL	EOCM	DDM	DMA		Reserved						CTN	ADCON	
		rw	rw	rw	rw								rw	rw	

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	SWRCST	Softw are start on regular channel. Setting 1 on this bit starts a conversion of a group of regular channels. It is set by softw are and cleared by softw are or by hardw are after the conversion starts.
29:28	ETMRC[1:0]	External trigger mode for regular channel 00: External trigger for regular channel disable 01: Rising edge of external trigger for regular channel enable 01: Falling edge of external trigger for regular channel enable 11: Rising and falling edge of external trigger for regular channel enable
27:24	ETSRC[3:0]	External trigger select for regular channel 0000: TIMER0 CH0 0001: TIMER0 CH1 0010: TIMER0 CH2 0011: TIMER1 CH1 0100: TIMER1 CH2 0101: TIMER1 CH3 0110: TIMER1 TRGO 0111: TIMER2 CH0 1000: TIMER2 TRGO 1001: TIMER3 CH3 1010: TIMER4 CH0 1011: TIMER4 CH1

		1100: TIMER4 CH2 1101: Reserved 1110: Reserved 1111: EXTI line 11
23	Reserved	Must be kept at reset value.
22	SWICST	Softw are start on inserted channel. Setting 1 on this bit starts a conversion of a group of inserted channels. It is set by softw are and cleared by softw are or by hardw are after the conversion starts.
21:20	ETMIC[1:0]	External trigger mode for inserted channel 00: External trigger for inserted channel disable 01: Rising edge of external trigger for inserted channel enable 01: Falling edge of external trigger for inserted channel enable 11: Rising and falling edge of external trigger for inserted channel enable
19:16	ETSIC[3:0]	External trigger select for inserted channel 0000: TIMER0 CH3 0001: TIMER0 TRGO 0010: TIMER1 CH0 0011: TIMER1 TRGO 0100: TIMER2 CH1 0101: TIMER2 CH3 0110: TIMER3 CH0 0111: TIMER3 CH1 1000: TIMER3 CH2 1001: TIMER3 TRGO 1010: TIMER4 CH3 1011: TIMER4 TRGO 1100: Reserved 1101: Reserved 1110: Reserved 1111: EXTI line 15
15:12	Reserved	Must be kept at reset value.
11	DAL	Data alignment 0: right alignment 1: left alignment
10	EOCM	End of conversion mode 0: Only at the end of a sequence of regular conversions, the EOC bit is set. Overflow detection is disabled unless DMA=1. 1: At the end of each regular conversion, the EOC bit is set. Overflow is detected automatically
9	DDM	DMA disable mode

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 DMA=1, the DMA engine issues a request at end of each regular conversion.

8	DMA	DMA request enable for regular channel. 0: DMA request disable 1: DMA request enable
7:2	Reserved	Must be kept at reset value.
1	CTN	Continuous mode 0: Continuous mode disable 1: Continuous mode enable
0	ADCON	ADC ON. The ADC will be waked up when this bit is changed from low to high and take a stabilization time. 0: ADC disable and power down 1: ADC enable

14.5.4. Sample time register 0 (ADC_SAMPT0)

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
Reserved										SPT11[2:0]		SPT10[2:0]			
										rw		rw			

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value
5:3	SPT11[2:0]	Refer to SPT10[2:0] description
2:0	SPT10[2:0]	Channel sampling time 000: 1.5 cycles 001: 14.5 cycles 010: 27.5 cycles 011: 55.5 cycles 100: 83.5 cycles

101: 111.5 cycles

110: 143.5 cycles

111: 479.5 cycles

14.5.5. Sample time register 1 (ADC_SAMPT1)

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		SPT9[2:0]		SPT8[2:0]		SPT7[2:0]		SPT6[2:0]		SPT5[2:1]					
		rw		rw		rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]		SPT4[2:0]		SPT3[2:0]		SPT2[2:0]		SPT1[2:0]		SPT0[2:0]					
		rw		rw		rw									

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:27	SPT9[2:0]	Refer to SPT0[2:0] description
26:24	SPT8[2:0]	Refer to SPT0[2:0] description
23:21	SPT7[2:0]	Refer to SPT0[2:0] description
20:18	SPT6[2:0]	Refer to SPT0[2:0] description
17:15	SPT5[2:0]	Refer to SPT0[2:0] description
14:12	SPT4[2:0]	Refer to SPT0[2:0] description
11:9	SPT3[2:0]	Refer to SPT0[2:0] description
8:6	SPT2[2:0]	Refer to SPT0[2:0] description
5:3	SPT1[2:0]	Refer to SPT0[2:0] description
2:0	SPT0[2:0]	Channel sampling time 000: 1.5 cycles 001: 14.5 cycles 010: 27.5 cycles 011: 55.5 cycles 100: 83.5 cycles 101: 111.5 cycles 110: 143.5 cycles 111: 479.5 cycles

14.5.6. Inserted channel data offset register x (ADC_IOFFx) (x=0..3)

Address offset: 0x14 + 0x04 * x (x=0..3)

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					IOFF [11:0]										rw	

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	IOFF[11:0]	Data offset for inserted channel x. These bits will be subtracted from the raw converted data when converting inserted channels. The conversion result can be read from the ADC_IDATAx registers.

14.5.7. Watchdog high threshold register (ADC_WDHT)

Address offset: 0x24

Reset value: 0x0000 0FFF

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					WDHT [11:0]										rw	

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WDHT[11:0]	Analog watchdog high threshold These bits define the high threshold for the analog watchdog.

14.5.8. Watchdog low threshold register (ADC_WDLT)

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					WDLT [11:0]										rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WDLT[11:0]	Analog watchdog low threshold These bits define the low threshold for the analog watchdog.

14.5.9. Regular sequence register 0 (ADC_RSQ0)

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Regular channel group length The total number of conversion in regular group equals to RL[3:0] +1.
19:0	Reserved	Must be kept at reset value.

14.5.10. Regular sequence register 1 (ADC_RSQ1)

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSQ8[4:0]				RSQ7[4:0]				RSQ6[4:0]						
	rw					rw				rw					

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ6[4:0]	Refer to RSQ0[4:0] description

14.5.11. Regular sequence register 2 (ADC_RSQ2)

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
Reserved	RSQ5[4:0]				RSQ4[4:0]				RSQ3[4:1]						
	rw					rw				rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ3[0]	RSQ2[4:0]				RSQ1[4:0]				RSQ0[4:0]						
	rw					rw				rw					

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ4[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ3[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ2[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ1[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ0[4:0]	The channel number (0..11) is written to these bits to select a channel as the nth conversion in the regular channel group.

14.5.12. Inserted sequence register (ADC_ISQ)

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										IL[1:0]	ISQ3[4:1]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISQ3[0]	ISQ2[4:0]				ISQ1[4:0]				ISQ0[4:0]						
rw									rw						rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	IL[1:0]	Inserted channel group length. The total number of conversion in inserted group equals IL[1:0] + 1.
19:15	ISQ3[4:0]	Refer to ISQ0[4:0] description
14:10	ISQ2[4:0]	Refer to ISQ0[4:0] description
9:5	ISQ1[4:0]	Refer to ISQ0[4:0] description
4:0	ISQ0[4:0]	The channel number (0..11) is written to these bits to select a channel as the nth conversion in the inserted channel group. Different from the regular conversion sequence, the inserted channels are converted starting from (4-IL [1:0]-1), if IL [1:0] length is less than 4. IL Insert channel order 11 ISQ0>>ISQ1>>ISQ2>>ISQ3 10 ISQ1>>ISQ2>>ISQ3 01 ISQ2>>ISQ3 00 ISQ3

14.5.13. Inserted data register x (ADC_IDATAx) (x=0..3)

Address offset: 0x3C + 0x04 * x (x=0..3)

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

IDATAn[15:0]

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	IDATAn[15:0]	Inserted number n conversion data These bits contain the number n conversion result, which is read only..

14.5.14. Regular data register (ADC_RDATA)

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

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RDATA[15:0]	Regular channel data These bits contain the conversion result from regular channel, which is read only.

14.5.15. Oversampling control register (ADC_OVSAMPCTL)

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

Reserved

rw

OVSS[3:0]

rw

OVSR[2:0]

rw

Reserved OVSEN

rw

Bits	Fields	Descriptions
-------------	---------------	---------------------

31:10	Reserved	Must be kept at reset value.
9	TOVS	<p>Triggered Oversampling This bit is set and cleared by software.</p> <p>0: All oversampled conversions for a channel are done consecutively after a trigger 1: Each conversion needs a trigger for a oversampled channel and the number of triggers is determined by the oversampling ratio(OVSR[2:0])</p> <p>Note: Software is allowed to write this bit only when ADCON=0 (which ensures that no conversion is ongoing).</p>
8:5	OVSS [3:0]	<p>Oversampling shift These bits are set and cleared by software.</p> <ul style="list-style-type: none"> 0000: No shift 0001: Shift 1 bit 0010: Shift 2 bits 0011: Shift 3 bits 0100: Shift 4 bits 0101: Shift 5 bits 0110: Shift 6 bits 0111: Shift 7 bits 1000: Shift 8 bits Other: reserved <p>Note: Software is allowed to write this bit only when ADCON =0 (which ensures that no conversion is ongoing).</p>
4:2	OVSR [2:0]	<p>Oversampling ratio This bit field defines the number of oversampling ratio.</p> <ul style="list-style-type: none"> 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x <p>Note: Software is allowed to write this bit only when ADCON =0 (which ensures that no conversion is ongoing).</p>
1	Reserved	Must be kept at reset value.
0	OVSEN	<p>Oversampling enable This bit is set and cleared by software.</p> <ul style="list-style-type: none"> 0: Oversampling disabled 1: Oversampling enabled <p>Note: Software is allowed to write this bit only when ADCON =0 (which ensures that no conversion is ongoing).</p>

14.5.16. Common control register (ADC_CCTL)

Address offset: 0x304

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	TSVREN	Channel 9 (temperature sensor) and 10 (internal reference voltage) enable of ADC. 0: Channel 9 and 10 of ADC disable 1: Channel 9 and 10 of ADC enable
22	VBATEN	Channel 11 (1/4 voltage of external battery) enable of ADC. 0: Channel 11 of ADC disable 1: Channel 11 of ADC enable
21:19	Reserved	Must be kept at reset value.
18:16	ADCCK[2:0]	ADC clock These bits configure the ADC clock. 000: PCLK2 / 2 001: PCLK2 / 4 010: PCLK2 / 6 011: PCLK2 / 8 100: HCLK / 5 101: HCLK / 6 110: HCLK / 10 111: HCLK / 20
15:0	Reserved	Must be kept at reset value.

15. 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 (WWDT). They offer a combination of a high safety level, flexibility of use and timing accuracy. Both watchdog timers are offered to resolve malfunctions of software.

The watchdog timer will generate a reset (or an interrupt in window watchdog timer) when the internal counter reaches a given value. The watchdog timer counter can be stopped while the processor is in the debug mode.

15.1. Free watchdog timer (FWDGT)

15.1.1. 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 causes a reset when the internal down counter reaches 0. The register write protection function in free watchdog timer can be enabled to prevent it from changing the configuration unexpectedly.

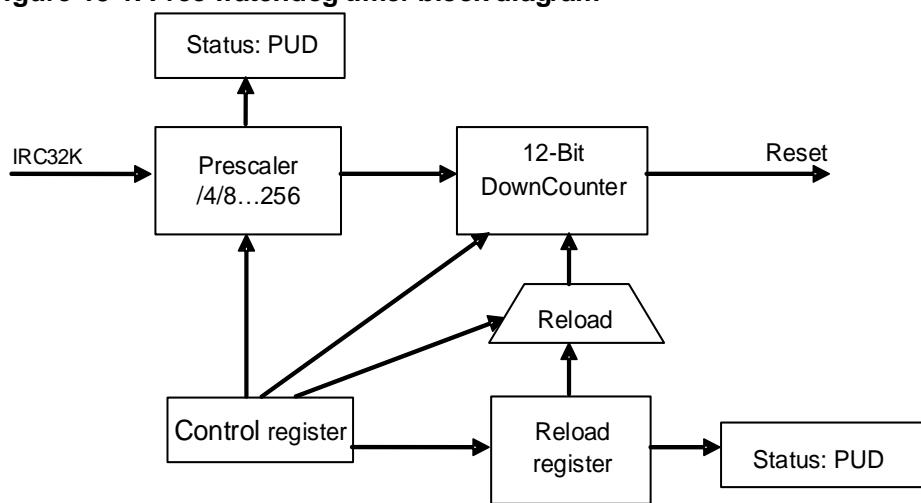
15.1.2. Characteristics

- Free-running 12-bit downcounter
- Reset when the downcounter reaches 0, if the watchdog is enabled
- Free clock source, FWDGT can operate even if the main clock fails such as in standby and Deep-sleep modes
- Hardware free watchdog timer bit, automatically start the FWDGT at power on
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode

15.1.3. Function overview

The free watchdog timer consists of an 8-stage prescaler and a 12-bit down-counter. [Figure 15-1. Free watchdog timer block diagram](#) shows the functional block of the free watchdog timer module.

Figure 15-1. Free watchdog timer block diagram



The free watchdog timer 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.

The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at anytime. The reload value comes from the FWDGT_RLD register. The software can prevent the watchdog reset by reloading the counter before the counter reaches the value 0x000.

The free watchdog timer can automatically start when power on if the hardware free watchdog timer bit in the device option bits is set. To avoid reset, the software should reload the counter before the counter reaches 0x000.

The FWDGT_PSC register and the FWDGT_RLD 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) or the reload value register (FWDGT_RLD) 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 15-1. 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	0.03125	511.90625
1/8	001	0.03125	1023.78125
1/16	010	0.03125	2047.53125
1/32	011	0.03125	4095.03125
1/64	100	0.03125	8190.03125
1/128	101	0.03125	16380.03125
1/256	110 or 111	0.03125	32760.03125

The FWDGT timeout can be more accurate by calibrating the IRC32K.

15.1.4. Register definition

FWDGT Secure access base address: 0x5000 3000

FWDGT Non-Secure access 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).

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

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 timer counter. When the counter reduces to 0, the free watchdog timer generates a reset 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).

31	30	29	28	27	26	25	24	23	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															
PSC[2:0]															
rw															

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 in 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 is reset before changing the prescaler value. However, after updating the prescaler value it is not necessary to wait until PUD is reset before continuing code execution.

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).

31	30	29	28	27	26	25	24	23	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				RLD [11:0]											
rw															

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	<p>Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT counter with the RLD value.</p> <p>These bits are write-protected. Write 0x5555 in 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.</p> <p>If several reload values are used by the application, it is mandatory to wait until RUD bit is reset before changing the reload value. However, after updating the reload value it is not necessary to wait until RUD is reset before continuing code execution.</p>

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).

31	30	29	28	27	26	25	24	23	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														RUD	PUD

r r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
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 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 FWDGT_PSC register.

15.2. Window watchdog timer (WWDT)

15.2.1. Overview

The window watchdog timer (WWDT) 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 0x3F (the CNT[6] bit has been cleared). The watchdog timer also causes a reset when the counter is refreshed before the counter reached the window register value. 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 0x40 or refreshes before the counter reaches the window value. Interrupt occurs if it is enabled.

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.

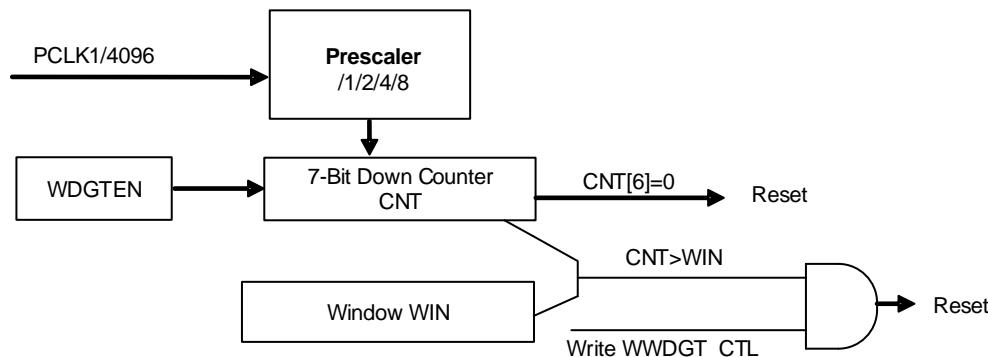
15.2.2. Characteristics

- Programmable free-running 7-bit down counter.
- Generate reset in two conditions when WWDT is enabled:
 - Reset when the counter reached 0x3F.
 - The counter is refreshed when the value of the counter is greater than the window register value.
- Early wakeup interrupt (EWI): if the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40 or refreshes before it reaches the window value.
- WWDT debug mode, the WWDT can stop or continue to work in debug mode.

15.2.3. Function overview

If the window watchdog timer is enabled (set the WDGTE bit in the WWDT_CTL), the watchdog timer cause a reset when the counter reaches 0x3F (the CNT[6] bit has been cleared), or the counter is refreshed before the counter reaches the window register value.

Figure 15-2. Window watchdog timer block diagram



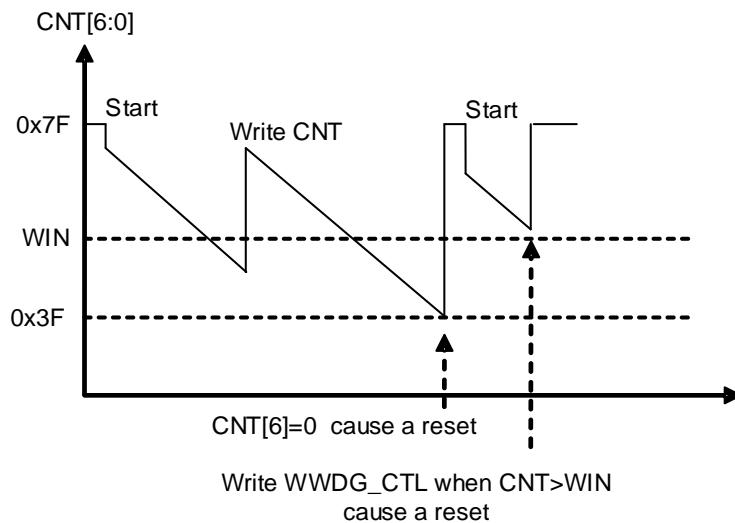
The watchdog is always disabled after power on reset. The software starts the watchdog by setting the WDGTE bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies that the CNT[6] bit should be set). The CNT[5:0] determine the maximum time interval between two reloading. The count down speed depends on the APB1 clock and the prescaler (PSC[1:0] bits in the WWDGT_CFG register).

The WIN[6:0] bits in the configuration register (WWDGT_CFG) specifies the window value. The software can prevent the reset event by reloading the down counter. The counter value is less than the window value and greater than 0x3F, otherwise the watchdog causes a reset.

The early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, and the interrupt will be generated when the counter reaches 0x40 or the counter is refreshed before it reaches the window value. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse 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 is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 15-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{\text{WWDGT}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PSC}} \times (\text{CNT}[5:0]+1) \quad (\text{ms}) \quad (13-1)$$

where:

t_{WWDGT} : WWDGT timeout

t_{PCLK1} : APB1 clock period measured in ms

The table below shows the minimum and maximum values of the t_{WWDGT} .

Table 15-2. Min-max timeout value at 45 MHz (f_{PCLK1})

Prescaler divider	PSC[1:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1/1	00	91.02 μs	5.83 ms
1/2	01	182.04 μs	11.65 ms
1/4	10	364.08 μs	23.30 ms
1/8	11	728.18 μs	46.60 ms

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.

15.2.4. Register definition

WWDGT Secure access base address: 0x5000 2C00

WWDG Non-Secure access base address: 0x4000 2C00

Control register (WWDGTR_CTL)

Address offset: 0x00

Reset value: 0x0000 007F

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	WDGTEN	<p>Start the Window watchdog timer. Cleared by a hardware reset. Writing 0 has no effect.</p> <p>0: Window watchdog timer disabled 1: Window watchdog timer enabled</p>
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occurs when the value of this counter decreases from 0x40 to 0x3F. When the value of this counter is greater than the window value, writing this counter also causes a reset.

Configuration register (WWDGT_CFG)

Address offset: 0x04

Reset value: 0x0000 007F

This register can be accessed by half-word(16-bit) or word(32-bit)

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.

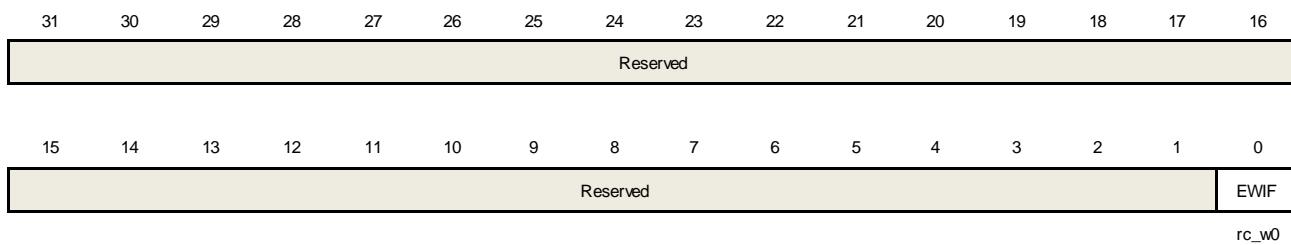
9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40. It can be cleared by a hardware reset or software clock reset (refer to 错误!未找到引用源。). A write operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. The time base of the watchdog counter 00: (PCLK1 / 4096) / 1 01: (PCLK1 / 4096) / 2 10: (PCLK1 / 4096) / 4 11: (PCLK1 / 4096) / 8
6:0	WIN[6:0]	The Window value. A reset occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the Window value.

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
31:1	Reserved	Must be kept at reset value.
0	EWIF	Early wakeup interrupt flag. When the counter reaches 0x40 or refreshes before it reaches the window value, this bit is set by hardware even the interrupt is not enabled (EWIE in WWDGT_CFG is cleared). This bit is cleared by writing 0. There is no effect when writing 1.

16. Real time clock (RTC)

16.1. Overview

The RTC provides a time which includes hour/minute/second/sub-second and a calendar includes year/month/day/week day. The time and calendar are expressed in BCD code except sub-second. Sub-second is expressed in binary code. Hour adjust for daylight saving time. Working in power saving mode and smart wakeup is software configurable. Support improving the calendar accuracy using extern accurate low frequency clock.

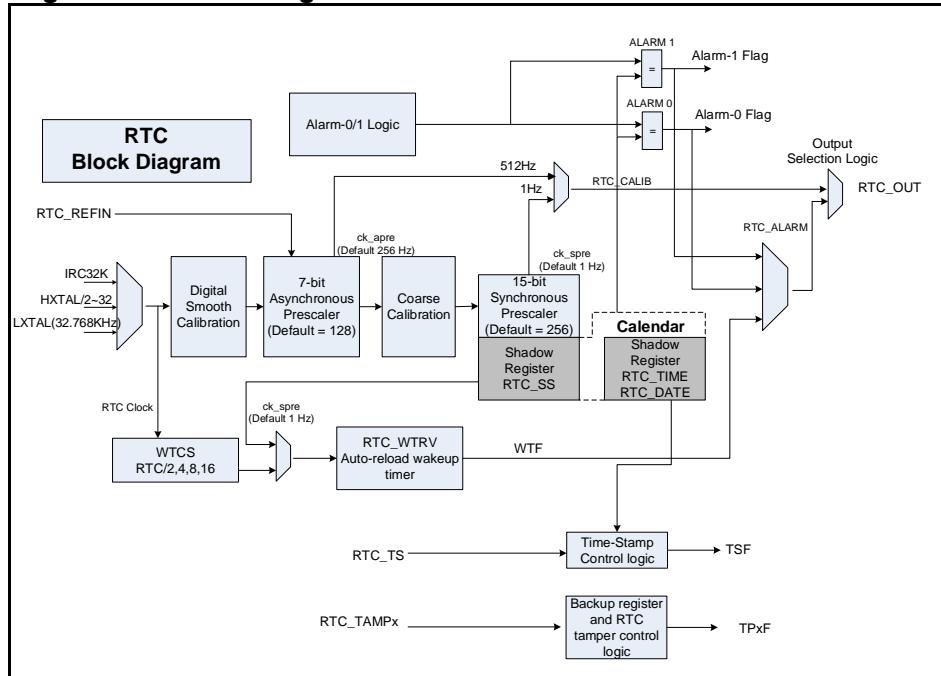
16.2. Characteristics

- Daylight saving compensation supported by software
- External high-accurate low frequency(50Hz or 60Hz) clock used to achieve higher calendar accuracy performed by reference clock detection option function
- Atomic clock adjust(max adjust accuracy is 0.95PPM) for calendar calibration performed by digital calibration function
- Sub-second adjustment by shift function
- Time-stamp function for saving event time
- Two Tamper sources can be chosen and tamper type is configurable
- Programmable calendar and two field maskable alarms
- Maskable interrupt source:
 - Alarm 0 and Alarm 1
 - Time-stamp detection
 - Tamper detection
 - Auto wakeup event
- Twenty 32-bit (80 bytes total) universal backup registers which can keep data under power saving mode. Backup register will be reset if tamper event detected
- Support TrustZone security protection
 - RTC global securable
 - The TZEN option bit is used to enable trustzone in the device. When TZEN = 1, trustzone is enable. When TZEN = 0, TrustZone is disabled, and at this time, the APB access to the RTC registers are non-secure.
 - Alarm 0, Alarm 1, wake-up Timer and timestamp can be independently configured as secure or non-secure
 - Tamper secure or non-secure configuration.
 - Configure the backup domain registers in three configurable-size areas. There are read/write secure area, write secure/read non-secure area and read/write non-secure area.

16.3. Function overview

16.3.1. Block diagram

Figure 16-1. Block diagram of RTC



The RTC unit includes:

- Alarm event/interrupt
- Tamper event/interrupt
- 32-bit backup registers
- Optional RTC output function:
 - 512Hz (default prescale) : PC15/PA3/PA8
 - 1Hz(default prescale): PC15/PA3/PA8
 - Alarm event(polarity is configurable): PC15/PA3/PA8
 - Automatic wakeup event(polarity is configurable): PC15/PA3/PA8
- Optional RTC input function:
 - Time stamp event detection(RTC_TS): PC15
 - Tamper 0 event detection(RTC_TAMP0): PC15
 - Tamper 1 event detection(RTC_TAMP0): PA2
 - Reference clock input RTC_REFIN(50 or 60 Hz)

It is possible to output RTC_OUT on PA3 or PA8 pin thanks to OUT2EN bit in RTC_CTL[31].

This output is not available in VBAT mode.

In addition, the TAMPER 1 alternate function corresponds to PA2 pin, This output is not available in VBAT mode.

16.3.2. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC32K and HXTAL with divided by 2~31(configured in RCU_CFG register).

In the RTC unit, there are two prescalers used for implementing the calendar and other functions. One prescaler is a 7-bit asynchronous prescaler and the other is a 15-bit synchronous prescaler. Asynchronous prescaler is mainly used for reducing power consumption. The asynchronous prescaler is recommended to set as high as possible if both prescalers are used.

The frequency formula of two prescalers is shown as below:

$$f_{ck_apre} = \frac{f_{rtcclk}}{\text{FACTOR_A} + 1} \quad (17-1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{\text{FACTOR_S} + 1} = \frac{f_{rtcclk}}{(\text{FACTOR_A} + 1) * (\text{FACTOR_S} + 1)} \quad (17-2)$$

The ck_apre clock is used to driven the RTC_SS down counter which stands for the time left to next second in binary format and when it reaches 0 it will automatically reload FACTOR_S value. The ck_spre clock is used to driven the calendar registers. Each clock will make second plus one.

16.3.3. Shadow registers introduction

BPSHAD control bit decides the location when APB bus accesses the RTC calendar register RTC_DATE, RTC_TIME and RTC_SS. By default, the BPSHAD is cleared, and APB bus accesses the shadow calendar registers. Shadow calendar registers is updated with the value of real calendar registers every two RTC clock and at the same time RSYNF bit will be set once. This update mechanism is not performed in Deep-Sleep mode and Standby mode. When exiting these modes, software must clear RSYNF bit and wait it is asserted (the max wait time is 2 RTC clock) before reading calendar register under BPSHAD=0 situation.

Note: When reading calendar registers (RTC_SS, RTC_TIME, RTC_DATE) under BPSHAD=0, the frequency of the APB clock (fapb) must be at least 7 times the frequency of the RTC clock (frtcclk).

System reset will reset the shadow calendar registers.

16.3.4. Configurable and field maskable alarm

RTC alarm function is divided into some fields and each has a maskable bit.

RTC alarm function can be enabled or disabled by ALRMxEN bit in RTC_CTL. If all the alarm fields value match the corresponding calendar value when ALRMxEN=1, the Alarm flag will be set.

Note: FACTOR_S in the RTC_PSC register must be larger than 3 if MSKS bit reset in RTC_ALRMxTD.

If a field is masked, the field is considered as matched in logic. If all the fields have been masked, the Alarm Flag will assert 3 RTC clock later after ALRM_xEN is set.

16.3.5. Configurable periodic auto-wakeup counter

In the RTC block, there is a 16-bit down counter designed to generate periodic wakeup flag.

This function is enabled by set the WTEN to 1 and can be running in power saving mode.

Two clock sources can be chose for the down counter:

- 1) RTC clock divided by 2/4/8/16

Assume RTC clock comes from LXTAL (32.768 KHz), this can periodically assert wakeup interrupt from 122us to 32s under the resolution down to 61us.

- 2) Internal clock ck_spre

Assume ck_spre is 1Hz, this can periodically assert wakeup interrupt from 1s to 36 hours under the resolution down to 1s.

- WTCS[2:1] = 0b10. This will make period to be 1s to 18 hours
- WTCS[2:1] = 0b11. This will make period to be 18 to 36 hours

When this function is enabled, the down counter is running. When it reaches 0, the WTF flag is set and the wakeup counter is automatically reloaded with RTC_WUT value.

When WTF asserts, software must then clear it.

If WTIE is set and this counter reaches 0, a wakeup interrupt will make system exit from the power saving mode. System reset has no influence on this function.

16.3.6. RTC initialization and configuration

RTC register write protection

BKPWEN bit in the PMU_CTL register is cleared in default, so writing to RTC registers needs setting BKPWEN bit ahead of time.

After power-on reset, most of RTC registers are write protected. Unlocking this protection is the first step before writing to them.

Following below steps will unlock the write protection:

1. Write '0xCA' into the RTC_WPK register
2. Write '0x53' into the RTC_WPK register

Writing a wrong value to RTC_WPK will make write protection valid again.

After Backup domain reset, some of the RTC registers are write-protected: RTC_TIME, RTC_DATE, RTC_PSC, RTC_COSC, RTC_HRFC, RTC_SHIFTCTL, the bit INITM in RTC_ICSR and the bits CS, S1H,A1H, REFEN in RTC_CTL.

The registers protected by INITPRIV are write-protected by the INIT KEY.

The registers protected by INITPRIP are write-protected by the INIT KEY.

The registers protected by CALDPRIV are write-protected by the CAL KEY.

In case RTCPRIP or INITPRIP is set in the RTC_PPM_CTL, and/or DPROT or INITSECP is cleared in the RTC_SPM_CTL: the INIT KEY is unlocked and locked only if the write accesses into the RTC_WPK register are done in the privilege and security mode defined by RTCPRIP, INITPRIP, DPROT, INITSECP configuration.

In case RTCPRIP or CALCPRIP is set in the RTC_PPM_CTL, and/or DPROT or CALSECP is cleared in the RTC_SPM_CTL: the CAL KEY is unlocked and locked only if the write accesses into the RTC_WPK register are done in the privilege and security mode defined by RTCPRIP, CALCPRIP, DPROT, CALSECP configuration.

Calendar initialization and configuration

The prescaler and calendar value can be programmed by the following steps:

1. Enter initialization mode (by setting INITM=1) and polling INITF bit until INITF=1.
2. Program both the asynchronous and synchronous prescaler factors in RTC_PSC register.
3. Write the initial calendar values into the shadow calendar registers (RTC_TIME and RTC_DATE), and use the CS bit in the RTC_CTL register to configure the time format (12 or 24 hours).
4. Exit the initialization mode (by setting INITM=0).

About 4 RTC clock cycles later, real calendar registers will load from shadow registers and calendar counter restarts.

Note: Reading calendar register (BPSHAD=0) after initialization, software should confirm the RSYNF bit to 1.

YCM flag indicates whether the calendar has been initialized by checking the year field of calendar.

Daylight saving Time

RTC unit supports daylight saving time adjustment through S1H, A1H and DSM bit.

S1H and A1H can subtract or add 1 hour to the calendar when the calendar is running. S1H and A1H operation can be tautologically set and DSM bit can be used to recording this adjust operation. After setting the S1H/A1H, subtract/add 1 hour will perform when next second comes.

Alarm function operation process

To avoid unexpected alarm assertion and metastable state, alarm function has an operation flow:

1. Disable Alarm (by resetting ALRMxEN in RTC_CTL)

-
2. Set the Alarm registers needed(RTC_ALRMxTD/RTC_ALRMxSS)
 3. Enable Alarm function (by setting ALRMxEN in the RTC_CTL)

16.3.7. Calendar reading

Reading calendar registers under BPSHAD=0

When BPSHAD=0, calendar value is read from shadow registers. For the existence of synchronization mechanism, a basic request has to meet: the APB1 bus clock frequency must be equal to or greater than 7 times the RTC clock frequency. APB1 bus clock frequency lower than RTC clock frequency is not allowed in any case whatever happens.

When APB1 bus clock frequency is not equal to or greater than 7 times the RTC clock frequency, the calendar reading flow should be obeyed:

1. Reading calendar time register and date register twice
2. If the two values are equal, the value can be seen as the correct value
3. If the two values are not equal, a third reading should be performed
4. The third value can be seen as the correct value

RSYNF is asserted once every 2 RTC clock and at this time point, the shadow registers will be updated to current time and date.

To ensure consistency of the 3 values (RTC_SS, RTC_TIME, and RTC_DATE), below consistency mechanism is used in hardware:

1. Reading RTC_SS will lock the updating of RTC_TIME and RTC_DATE
2. Reading RTC_TIME will lock the updating of RTC_DATE
3. Reading RTC_DATE will unlock updating of RTC_TIME and RTC_DATE

If the software wants to read calendar in a short time interval (smaller than 2 RTCCLK periods), RSYNF must be cleared by software after the first calendar read, and then the software must wait until RSYNF is set again before next reading.

In below situations, software should wait RSYNF bit asserted before reading calendar registers (RTC_SS, RTC_TIME, and RTC_DATE):

1. After a system reset
2. After an initialization
3. After shift function

Especially that software must clear RSYNF bit and wait it asserted before reading calendar register after wakeup from power saving mode.

Reading calendar registers under BPSHAD=1

When BPSHAD=1, RSYNF is cleared and maintains as 0 by hardware so reading calendar registers does not care about RSYNF bit. Current calendar value is read from real-time

calendar counter directly. The benefit of this configuration is that software can get the real current time without any delay after wakeup from power saving mode (Deep-sleep /Standby Mode).

Because of no RSYNF bit periodic assertion, the results of the different calendar registers (RTC_SS/RTC_TIME/RTC_DATE) might not be coherent with each other when clock ck_apre edge occurs between two reading calendar registers.

In addition, if current calendar register is changing and at the same time the APB bus reading calendar register is also performing, the value of the calendar register read out might be not correct.

To ensure the correctness and consistency of the calendar value, software must perform reading operation as this: read all calendar registers continuously, if the last two values are the same, the data is coherent and correct.

16.3.8. Resetting the RTC

There are two reset sources used in RTC unit: system reset and backup domain reset.

System reset will affect calendar shadow registers and some bits of the RTC_STAT. When system reset is valid, the bits or registers mentioned before are reset to the default value.

Backup domain reset will affect the following registers and system reset will not affect them:

- RTC current real-time calendar registers
- RTC Control register (RTC_CTL)
- RTC Prescaler register (RTC_PSC)
- RTC Wakeup timer register (RTC_WUT)
- RTC Coarse calibration register (RTC_COSC)
- RTC High resolution frequency compensation register (RTC_HRFC)
- RTC Shift control register (RTC_SHIFTCTL)
- RTC Time stamp registers (RTC_SSTS/RTC_TTS/RTC_DTS)
- RTC Tamper register (RTC_TAMP)
- RTC Backup registers (RTC_BKPx)
- RTC Alarm registers (RTC_ALRMxSS/RTC_ALRMxTD)

The RTC unit will go on running when system reset occurs or enter power saving mode, but if backup domain reset occurs, RTC will stop counting and all registers will reset.

16.3.9. RTC shift function

When there is a remote clock with higher degree of precision and RTC 1Hz clock (ck_spre) has an offset (in a fraction of a second) with the remote clock, RTC unit provides a function named shift function to remove this offset and thus make second precision higher.

RTC_SS register indicates the fraction of a second in binary format and is down counting when RTC is running. Therefore by adding the SFS[14:0] value to the synchronous prescaler

counter SSC[15:0] or by adding the SFS[14:0] value to the synchronous prescaler counter SSC[15:0] and at the same time set A1S bit can delay or advance the time when next second arrives.

The maximal RTC_SS value depends on the FACTOR_S value in RTC_PSC. The higher FACTOR_S, the higher adjust precision.

Because of the 1Hz clock (ck_spre) is generated by FACTOR_A and FACTOR_S, the higher FACTOR_S means the lower FACTOR_A, then more power consuming.

Note: Before using shift function, the software must check the MSB of SSC in RTC_SS (SSC[15]) and confirm it is 0.

After writing RTC_SHIFTCTL register, the SOPF bit in RTC_STAT will be set at once. When shift operation is complete, SOPF bit is cleared by hardware. System reset does not affect SOPF bit.

Shift operation only works correctly when REFEN=0.

Software must not write to RTC_SHIFTCTL if REFEN=1.

16.3.10. RTC reference clock detection

RTC reference clock detection is another way to increase the precision of RTC second. To enable this function, you should have an external clock source (50Hz or 60 Hz) which is more precise than LXTAL clock source.

After enabling this function (REFEN=1), each 1Hz clock (ck_spre) edge is compared to the nearest RTC_REFIN clock edge. In most cases, the two clock edges are aligned every time. But when two clock edges are misaligned for the reason of LXTAL poor precision, the RTC reference clock detection function will shift the 1Hz clock edge a little to make next 1Hz clock edge aligned to reference clock edge.

When REFEN=1, a time window is applied at every second update time different detection state will use different window period.

7 ck_apre window is used when detecting the first reference clock edge and 3 ck_apre window is used for the edge aligned operation.

Whatever window used, the asynchronous prescaler counter will be forced to reload when the reference clock is detected in the window. When the two clock (ck_spre and reference clock) edges are aligned, this reload operation has no effect for 1Hz clock. But when the two clock edge are not aligned, this reload operation will shift ck_spre clock edge a bit to make the ck_spre(1Hz) clock edge aligned to the reference clock edge.

When reference detection function is running while the external reference clock is removed (no reference clock edge found in 3 ck_apre window), the calendar updating still can be performed by LXTAL clock only. If the reference clock is recovered later, detection function will use 7 ck_apre window to identify the reference clock and use 3 ck_apre window to adjust the 1Hz clock (ck_spre) edge.

Note: Software must configure the FACTOR_A=0x7F and FACTOR_S=0xFF before enabling reference detection function (REFEN=1)

Reference detection function does not work in Standby Mode and must not be used with coarse digital function.

16.3.11. RTC coarse digital calibration

There are two digital methods can be chose for calibration: coarse digital calibration and smooth digital calibration. These two types cannot be used together.

Coarse digital calibration can be used to add or mask ck_apre clock cycles at the output of the asynchronous prescaler.

When COSD=0, 2 ck_apre cycles are added every minute for the first 2xCOSS minutes. The effect of such configuration will make calendar to be updated sooner.

When COSD=1, 1 ck_apre cycle is removed every minute for the first 2xCOSS minutes. The effect of such configuration will make calendar to be updated later.

Only in initialization mode can configure coarse calibration and the function starts after clearing INITM bit. The full calibration window lasts 64 minutes. The first 2xCOSS minutes of this 64-minute window are take adjust.

About 2PPM resolution is taken for negative calibration and about 4PPM resolution is taken for positive calibration.

Note: The calibration can be performed either on LXTAL or HXTAL clock. If FACTOR_A<6, the calibration may not work correctly.

Example:

FACTOR_A and FACTOR_S are default values. LXTAL is the RTC clock source and frequency is 32.768 KHz.

During a calibration window (64 minutes), the ck_apre clock frequency is only adjusted in the first 2xCOSS minutes. If COSS=1, this means only the first 2 minutes of 64 minutes will make adjustment.

The calibration step therefore has the effect of adding 512 or subtracting 256 oscillator cycles for each calibration window (64min x 60s/min x 32768cycles/s). In another word, this is equivalent to +4.069PPM or -2.035PPM per calibration step. Then for one month running, the minimum calibration step is +10.5 or -5.27 seconds and the maximum calibration step is +5.45 to -2.72 minutes.

16.3.12. RTC smooth digital calibration

RTC smooth calibration function is a way to calibrate the RTC frequency based on RTC clock in a configurable period time.

This calibration is equally executed in a period time and the cycle number of the RTC clock in the period time will be added or subtracted. The resolution of the calibration is about 0.954PPM with the range from -487.1PPM to +488.5PPM.

The calibration period time can be configured to the 220/219/218 RTC clock cycles which stands for 32/16/8 seconds if RTC input frequency is 32.768 KHz.

The High resolution frequency compensation register (RTC_HRFC) specifies the number of RTCCLK clock cycles to be calibrated during the period time:

So using CMSK can mask clock cycles from 0 to 511 and thus the RTC frequency can be reduced by up to 487.1PPM.

To increase the RTC frequency the FREQI bit can be set. If FREQI bit is set, there will be 512 additional cycles to be added during period time which means every 211/210/29(32/16/8 seconds) RTC clock insert one cycle.

So using FREQI can increase the RTC frequency by 488.5PPM.

The combined using of CMSK and FREQI can adjust the RTC cycles from -511 to +512 cycles in the period time which means the calibration range is -487.1PPM to +488.5PPM with a resolution of about 0.954PPM.

When calibration function is running, the output frequency of calibration is calculated by the following formula:

$$f_{\text{cal}} = f_{\text{rtcclk}} \times \left(1 + \frac{FREQI \times 512 - CMSK}{2^N + CMSK - FREQI \times 512}\right) \quad (17.3)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Calibration when FACTOR_A<3

When asynchronous prescaler value (FACTOR_A) is set to less than 3, software should not set FREQI bit to 1 when using calibration function. FREQI setting will be ignored when FACTOR_A<3.

When the FACTOR_A is less than 3, the FACTOR_S value should be set to a value less than the nominal value. Assuming that RTC clock frequency is nominal 32.768 KHz, the corresponding FACTOR_S should be set as following rule:

FACTOR_A = 2: 2 less than nominal FACTOR_S (8189 with 32.768 KHz)

FACTOR_A = 1: 4 less than nominal FACTOR_S (16379 with 32.768 KHz)

FACTOR_A = 0: 8 less than nominal FACTOR_S (32759 with 32.768 KHz)

When the FACTOR_A is less than 3, CMSK is 0x100, the formula of calibration frequency is as follows:

$$f_{\text{cal}} = f_{\text{rtcclk}} \times \left(1 + \frac{256 - CMSK}{2^N + CMSK - 256}\right) \quad (17.4)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Verifying the RTC calibration

Calibration 1Hz output is provided to assist software to measure and verify the RTC precision.

Up to 2 RTC clock cycles measurement error may occur when measuring the RTC frequency over a limited measurement period. To eliminate this measurement error the measurement period should be the same as the calibration period.

- When the calibration period is 32 seconds(this is default configuration)

Using exactly 32s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.477PPM (0.5 RTCCLK cycles over 32s)

- When the calibration period is 16 seconds(by setting CWND16 bit)

In this configuration, CMSK[0] is fixed to 0 by hardware. Using exactly 16s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.954PPM (0.5 RTCCLK cycles over 16s)

- When the calibration period is 8 seconds(by setting CWND8 bit)

In this configuration, CMSK[1:0] is fixed to 0 by hardware. Using exactly 8s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 1.907PPM (0.5 RTCCLK cycles over 8s)

Re-calibration on-the-fly

When the INITF bit is 0, software can update the value of RTC_HRFC using following steps:

- 1) Wait the SCPF=0
- 2) Write the new value into RTC_HRFC register
- 3) After 3 ck_apre clocks, the new calibration settings take effect

16.3.13. Time-stamp function

Time-stamp function is performed on RTC_TS pin and is enabled by control bit TSEN.

When a time-stamp event occurs on RTC_TS pin, the calendar value will be saved in time-stamp registers (RTC_DTS/RTC_TTS/RTC_SSTS) and the time-stamp flag (TSF) is set to 1 by hardware. Time-stamp event can generate an interrupt if time-stamp interrupt enable (TSIE) is set.

Time-stamp registers only record the calendar at the first time time-stamp event occurs which means that time-stamp registers will not change when TSF=1.

To extend the time-stamp event source, one optional feature is provided: tamper function can also be considered as time-stamp function if TPTS is set.

Note: When the time-stamp event occurs, TSF is set 2 ck_apre cycles delay because of synchronization mechanism.

16.3.14. Tamper detection

The RTC_TAMPx pin input can be used for tamper event detection under edge detection mode or level detection mode with configurable filtering setting.

RTC backup registers (RTC_BKPx)

The RTC backup registers are located in the VDD backup domain that remains powered-on by V_{BAT} even if V_{DD} power is switched off. The wake up action from Standby Mode or System Reset does not affect these registers.

These registers are only reset by detected tamper event and backup domain reset.

Tamper detection function initialization

RTC tamper detection function can be independently enabled on tamper input pin by setting corresponding TPxEN bit. Tamper detection configuration is set before enable TPxEN bit. When the tamper event is detected, the corresponding flag (TPxF) will assert. Tamper event can generate an interrupt if tamper interrupt enable (TPIE) is set.

The backup registers are reset when a tamper detection event occurs except if the TAMPxNOER bit is set in the RTC_TAMP register. The backup registers and the device secrets erased by tamp_erase signal can be reset by software by setting the BKERASE bit in the RTC_TAMP register.

Timestamp on tamper event

The TPTS bit can control whether the tamper detection function is used as time-stamp function. If the bit is set to 1, the TSF bit will be set when the tamper event detected as if “enable” the time-stamp function. Whatever the TPTS bit is, the TPxF will assert when tamper event detected.

Edge detection mode on tamper input detection

When FLT bit is set to 0x0, the tamper detection is set to edge detection mode and TPxEG bit determines the rising edge or falling edge is the detecting edge. When tamper detection is under edge detection mode, the internal pull-up resistors on the tamper detection input pin are deactivated.

Because of detecting the tamper event will reset the backup registers (RTC_BKPx), writing to the backup register should ensure that the tamper event reset and the writing operation will not occur at the same time, a recommend way to avoid this situation is disable the tamper detection before writing to the backup register and re-enable tamper detection after finish writing.

Note: Tamper0 detection is still running when V_{DD} power is switched off if tamper is enabled.

Level detection mode with configurable filtering on tamper input detection

When FLT bit is not reset to 0x0, the tamper detection is set to level detection mode and FLT bit determines the consecutive number of samples (2, 4 or 8) needed for valid level. When DISPU is set to 0x0(this is default), the internal pull-up resistance will pre-charge the tamper input pin before each sampling and thus larger capacitance is allowed to connect to the tamper input pin. The pre-charge duration is configured through PRCH bit. Higher capacitance needs long pre-charge time.

The time interval between each sampling is also configurable. Through adjusting the sampling frequency (FREQ), software can balance between the power consuming and tamper detection latency.

16.3.15. Calibration clock output

Calibration clock can be output on the PC15/PA3/PA8 if COEN bit is set to 1.

When the COS bit is set to 0(this is default) and asynchronous prescaler is set to 0x7F(FACTOR_A), the frequency of RTC_CALIB is $f_{\text{rtcclk}}/64$. When the RTCCLK is 32.768KHz, RTC_CALIB output is corresponding to 512Hz. It's recommend to using rising edge of RTC_CALIB output for there may be a light jitter on falling edge.

When the COS bit is set to 1, the RTC_CALIB frequency is:

$$f_{\text{rtc_calib}} = \frac{f_{\text{rtcclk}}}{(\text{FACTOR}_A+1) \times (\text{FACTOR}_S+1)} \quad (17-5)$$

When the RTCCLK is 32.768 KHz, RTC_CALIB output is corresponding to 1Hz if prescaler are default values.

16.3.16. Alarm output

When OS control bits are not reset, RTC_ALARM alternate function output is enabled. This function will directly output the content of alarm flag or auto wakeup flag bit in RTC_STAT.

The OPOL bit in RTC_CTL can configure the polarity of the alarm or auto wakeup flag output which means that the RTC_ALARM output is the opposite of the corresponding flag bit or not.

16.3.17. RTC security protection

The rules for accessing RTC registers in secure and nonsecure protection modes are shown in [Table 16-1. RTC register secure access rules](#).

Table 16-1. RTC register secure access rules

Access mode	Read		Write	
	Secure access	Nonsecure access	Secure access	Nonsecure access
RTCSECP = 0	Allow ed(except for the backup registers)	Allow ed access RTC_SPM_CTL, RTC_PPM_CTL, RTC_NSMI_STAT, RTC_TIME, RTC_DATE, RTC_SS, RTC_PSC, RTC_COSC register	Allow ed(except for the backup registers)	Not allowed
RTCSECP = 1	By configuring the INITSECP, CALSECP, TSDSECP, WUTSECP, ALRM1SECP, TAMPSECP, ALRM0SECP bit in RTC_SPM_CTL register, refer to Table 16-2. RTC secure mode configuration summary for details.			

The summary of the RTC secured protected bits in RCU_SPM_CTL register is show as the [Table 16-2. RTC secure mode configuration summary](#).

Table 16-2. RTC secure mode configuration summary

Configuration bit in RTC_SPM_CTL	Write in secure mode	Read in secure mode	Read in non-secure mode
INITSECP=0	Allow ed access RTC_TIME, RTC_DATE and TC_PSC register; INITM in RTC_ICSR; CR control bits in RTC_CTL; INITSECP in RTC_SPM_CTL.	Allow ed access RTC_TIME, RTC_DATE and TC_PSC register; INITM in RTC_ICSR; CR control bits in RTC_CTL; INITSECP in RTC_SPM_CTL.	Allow ed access RTC_TIME, RTC_DATE and TC_PSC register; INITM in RTC_ICSR; CR control bits in RTC_CTL; INITSECP in RTC_SPM_CTL.
CALSECP=0	Allow ed access RTC_SHIFTCTL, RTC_HRFC and RTC_COSC registers; A1H, S1H and REFEN control bits in RTC_CTL; CALCSECP in the RTC_SPM_CTL.	Allow ed access RTC_SHIFTCTL, RTC_HRFC and RTC_COSC registers; A1H, S1H and REFEN control bits in RTC_CTL; CALCSECP in the RTC_SPM_CTL.	Allow ed access RTC_SHIFTCTL, RTC_HRFC and RTC_COSC registers; A1H, S1H and REFEN control bits in RTC_CTL; CALCSECP in the RTC_SPM_CTL.
ALRM0SECP=0	Allow ed access RTC_ALRM0TD, RTC_ALRM0SS registers; ALRM0EN, ALRM0IE in RTC_CTL;	Allow ed access RTC_ALRM0TD, RTC_ALRM0SS registers; ALRM0EN, ALRM0IE in	

Configuration bit in RTC_SPM_CTL	Write in secure mode	Read in secure mode	Read in non-secure mode
	ALRM0FC in RTC_STATC; ALRM0F in RTC_STAT; ALRM0SMF in RTC_SMI_STAT; ALRM0SECP in RTC_SPM_CTL.	RTC_CTL; ALRM0FC in the RTC_STATC; ALRM0F in RTC_STAT; ALRM0SMF in RTC_SMI_STAT.	
ALRM1SECP=0	Allow ed access RTC_ALRM1TD, RTC_ALRM1SS registers; ALRM1EN, ALRM1IE in RTC_CTL; ALRM1FC in RTC_STATC; ALRM1F in RTC_STAT; ALRM1MSF in RTC_SMI_STAT; ALRM1SECP in the RTC_SPM_CTL.	Allow ed access RTC_ALRM1TD, RTC_ALRM1SS registers; ALRM1EN, ALRM1IE in RTC_CTL; ALRM1FC in RTC_STATC; ALRM1F in RTC_STAT; ALRM1MSF in RTC_SMI_STAT.	
WUTSECP=0	Allow ed access RTC_WUT register; WTEN, WTIE and WTCS control bits in the RTC_CTL; WTFC in the RTC_STATC; WTF in RTC_STAT; WTSMF in RTC_SMI_STAT; WUTSECP in the RTC_SPM_CTL.	Allow ed access RTC_WUT register; WTEN, WTIE and WTCS control bits in the RTC_CTL; WTFC in the RTC_STATC; WTF in RTC_STAT; WTSMF in RTC_SMI_STAT.	
TSSECP=0	Allow ed access RTC_TTS, RTC_DTS and RTC_SSTS registers; TSEN, TSIE, TSEG control bits in the RTC_CTL; TSFC bits in the RTC_STATC; TSF, TSOVRF in RTC_STAT; TSMF, TSOVRSMF in RTC_SMI_STAT; TSSECP in the RTC_SPM_CTL.	Allow ed access RTC_TTS, RTC_DTS and RTC_SSTS registers; TSEN, TSIE, TSEG control bits in the RTC_CTL; TSFC bits in the RTC_STATC; TSF, TSOVRF in RTC_STAT; TSMF, TSOVRSMF in RTC_SMI_STAT.	

Configuration bit in RTC_SPM_CTL	Write in secure mode	Read in secure mode	Read in non-secure mode
TAMPSECP=0	Allow ed access RTC_TAMP register, except for the AOT bit; TPxF in RTC_STATC, TPxF in RTC_STAT, TPxMF in RTC_SMI_STAT; TAMPSECP in the RTC_SPM_CTL.	Allow ed access RTC_TAMP register, except for the AOT bit; TPxF in RTC_STATC, TPxF in RTC_STAT; TPxMF in RTC_SMI_STAT	

By default, after the backup domain is powered on and reset, except for the RTC_SPM_CTL register, all other RTC registers can be read and written in secure and non-secure modes.

Prohibit non-secure access to secure-protected registers, when non-secure access, no bus error will be generated. When the register is write-protected, the non-secure access to write register is invalid. When the register is read-protected, the non-secure access to read register value is 0. When the register is globally protected, TZIAC will generate a message of flag or interrupt. When only few bits of the register are protected, TZIAC will not generate any messages (except backup registers protection).

After the system is reset, the RTC protection related configuration unchanged. As long as one of the functions of the RTC is configured to be secured, the reset and clock control of the RTC are also secure in the RCU.

16.3.18. RTC privilege protection

The rules for accessing RTC registers in secure and nonsecure protection modes are shown in [Table 16-3. RTC register privilege access rules](#).

Table 16-3. RTC register privilege access rules

Access mode	Read		Write	
	Privilege access	Unprivilege access	Privilege access	Unprivilege access
RTCP RIP = 0	Allow ed(except for the backup registers)	Allow ed access RTC_SPM_CTL, RTC_PPM_CTL, RTC_TIME, RTC_DATE, RTC_SS, RTC_PSC and RTC_COSC register	Allow ed(except for the backup registers)	Not allowed
RTCP RIP = 1				By configuring the INITPRIP, CALCPRIP, TSPRIP, WUTPRIP, ALR1PRV, TAMPPRIP or ALRM0PRIP bit in RTC_PPM_CTL register, refer to Table 16-4. RTC privileged protected mode configuration summary for details.

The summary of the RTC privileged protected bits in RCU_PPM_CTL register is show as the [Table 16-4. RTC privileged protected mode configuration summary](#).

Table 16-4. RTC privileged protected mode configuration summary

Configuration bit in RTC_PPM_CTL	Write in privilege mode	Read in privilege mode	Read in non-privilege mode
INITPRIP =1	Allow ed access RTC_TIME, RTC_DATE, RTC_PSC registers; INITM in RTC_ICSR; CR control bits in RTC_CR; INITPRIP in RTC_PPM_CTL.	Allow ed access RTC_TIME, RTC_DATE, RTC_PSC registers; INITM in RTC_ICSR; CR control bits in RTC_CR; INITPRIP in RTC_PPM_CTL.	Allow ed access RTC_TIME, RTC_DATE, RTC_PSC registers; INITM in RTC_ICSR; CR control bits in RTC_CR; INITPRIP in RTC_PPM_CTL.
CALCPRIP =1	Allow ed access RTC_SHIFTCTL and RTC_COSC registers; A1H, S1H and REFEN control bits in the RTC_CTL; CALPRIP in the RTC_PPM_CTL.	Allow ed access RTC_SHIFTCTL and RTC_COSC registers; A1H, S1H and REFEN control bits in the RTC_CTL; CALPRIP in the RTC_PPM_CTL.	Allow ed access RTC_SHIFTCTL and RTC_COSC registers; A1H, S1H and REFEN control bits in the RTC_CTL; CALPRIP in the RTC_PPM_CTL.
ALRM0PRIP =1	Allow ed access RTC_ALRM0TD, RTC_ALRM0SS registers; ALRM0EN and ALRM0IE in RTC_CTL; ALRM0FC in RTC_STATC; ALRM0F in RTC_STAT; ALRM0NSMF in RTC_NSMI_STAT; ALRM0SMF in RTC_SMI_STAT; ALRM0PRIP in RTC_PPM_CTL.	Allow ed access RTC_ALRM0TD, RTC_ALRM0SS registers; ALRM0EN and ALRM0IE in RTC_CTL; ALRM0FC in RTC_STATC; ALRM0F in RTC_STAT; ALRM0NSMF in RTC_NSMI_STAT; ALRM0SMF in RTC_SMI_STAT.	
ALRM1PRIP =1	Allow ed access RTC_ALRM1TD, RTC_ALRM1SS registers; ALRM1EN and ALRM1IE in RTC_CTL; ALRM1FC in	Allow ed access RTC_ALRM1TD, RTC_ALRM1SS registers; ALRM1EN and ALRM1IE in RTC_CTL; ALRM1FC in	

Configuration bit in RTC_PPM_CTL	Write in privilege mode	Read in privilege mode	Read in non-privilege mode
	RTC_STATIC, ALRM1F in RTC_STAT, ALRM1NSMF in RTC_NSMI_STAT ; ALRM1SMF in RTC_SMI_STAT; ALRM1PRIP in RTC_PPM_CTL.	RTC_STATIC, ALRM1F in RTC_STAT, ALRM1NSMF in RTC_NSMI_STAT ; ALRM1SMF in RTC_SMI_STAT.	
WUTPRIP =1	Allow ed access RTC_WUT register; WTEN, WTIE and WTCS control bits in RTC_CTL; WTFC in RTC_STATC; WTF in RTC_STAT; WTNSMF in RTC_NSMI_STAT; WTSMF in RTC_SMI_STAT; ALRM1PRIP in RTC_PPM_CTL.	Allow ed access RTC_WUT register; WTEN, WTIE and WTCS control bits in RTC_CTL; WTFC in RTC_STATC; WTF in RTC_STAT; WTNSMF in RTC_NSMI_STAT; WTSMF in RTC_SMI_STAT.	
TSPRIP =1	Allow ed access RTC_TTS, RTC_DTS and RTC_SSTS registers; TSEN, TSIE, TSEG control bits in RTC_CTL; TSOVRFC and TSFC bits in RTC_STATC, TSF; TSOVRF in RTC_STAT; TSSMF, TSOVRSMF in RTC_SMI_STAT; TSPRIP in RTC_PPM_CTL.	Allow ed access RTC_TTS, RTC_DTS and RTC_SSTS registers; TSEN, TSIE, TSEG control bits in RTC_CTL; TSOVRFC and TSFC bits in RTC_STATC, TSF; TSOVRF in RTC_STAT; TSSMF, TSOVRSMF in RTC_SMI_STAT.	
TAMPPRIP =1	Allow ed access RTC_TAMP register, except for the AOT bit; TPxFC bits in the RTC_STATC; TPxF bits in RTC_STAT; TPxMF in RTC_SMI_STAT;	Allow ed access TC_TAMP register, except for the AOT bit; TPxFC bits in the RTC_STATC; TPxF bits in RTC_STAT; TPxMF in RTC_SMI_STAT.	

Configuration bit in RTC_PPM_CTL	Write in privilege mode	Read in privilege mode	Read in non-privilege mode
	TAMPPRIP in RTC_RTC_PPM_CTL.		

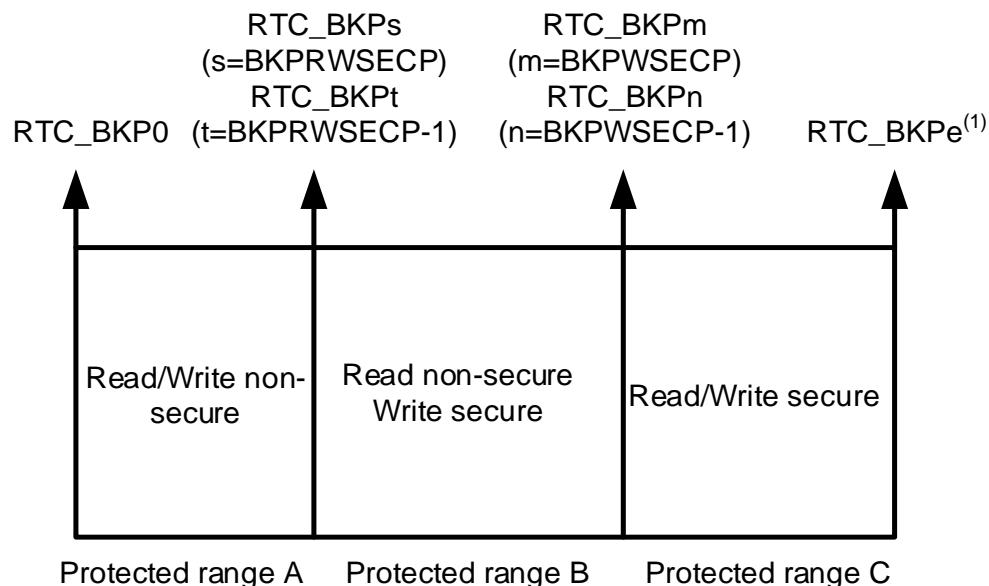
By default, after the backup domain is powered on and reset, except for the RTC_PPM_CTL register, all other RTC registers can be read and written in privilege and unprivilege modes. After the system is reset, the RTC protection related configuration unchanged.

Prohibit unprivileged access to privileged -protected registers, when unprivileged access, no bus error will be generated. When the register is write-protected, the unprivileged access to write register is invalid. When the register is read-protected, the the unprivileged access to read register value is 0.

16.3.19. RTC backup registers protection

The security protection status of the backup domain register is configured through bits BKPRWSECP[7:0] and BKPWSECP[7:0] in RTC_SPM_CTL register. Configure protection zone A through BKPRWSECP[7:0] and BKPRWPRIP, configure protection zone B through BKPRWSECP[7:0], BKPWSECP[7:0] and BKPWPRIP. The backup domain registers security protection configuration is shown in [Figure 16-2. Backup registers secure protections configuration](#).

Figure 16-2. Backup registers secure protections configuration



NOTE: (1) e represents the index value of the last backup register

When Trustzone is enabled, the BKPWPRIP and BKPRWPRIP bits in the RTC_PPM_CTL register can only be written via secure access. When Trustzone is disabled, BKPRWSECP[7:0] and BKPWSECP[7:0] in RTC_SPM_CTL register can be read and written

with non-secure access.

16.3.20. RTC power saving mode management

Table 16-5. RTC power saving mode management

Mode	Active in Mode	Exit Mode
Sleep	Yes	RTC Interrupts
Deep-Sleep	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Tamper Event / Timestamp Event / Wake up
Standby	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Tamper Event / Timestamp Event / Wake up

16.3.21. RTC interrupts

All RTC interrupts are connected to the EXTI controller. When RTC is in secure mode, configure the RTC_SMI_STAT register to set the interrupt channel, and configure the RTC_SMI_STAT register when RTC is in non-secure mode.

Below steps should be followed if you want to use the RTC alarm/tamper/timestamp/auto wakeup interrupt:

- 1) Configure and enable the corresponding interrupt line to RTC alarm/tamper/timestamp/auto wakeup event of EXTI and set the rising edge for triggering
- 2) Configure and enable the RTC alarm/tamper/timestamp/auto wakeup interrupt
- 3) Configure and enable the RTC alarm/tamper/timestamp/auto wakeup function

Table 16-6. RTC non-secure interrupts control

Interrupt	Event flag	Control bit	Clear interrupt flag	Exit sleep	Exit deep-sleep and standby
Alarm 0	ALRM0F	ALRM0IE and(ALRM0SECP=1 and RTCSECP=1)	w rite 1 in ALRM0FC	Y	Y(“)
Alarm 1	ALRM1F	ALRM1IE and(ALRM1SECP=1 and RTCSECP=1)	w rite 1 in ALRM1FC	Y	Y(“)
Wakeup	WTF	WTIE and(WUTSECP=1 and RTCSECP=1)	w rite 1 in WTFC	Y	Y(“)
Timestamp	TSF	TSIE and(TSSECP=1 and	w rite 1 in TSFC	Y	Y(“)

		RTCSECP=1)			
Tamper x	TPxF	TPxE and(TAMPSECP=1 and RTCSECP=1)	Write 1 in TPxF	Y	Y ^(*)

NOTE: (*)Only active when RTC clock source is LXTAL or IRC32K.

Table 16-7. RTC secure interrupts control

Interrupt	Event flag	Control bit	Clear interrupt flag	Exit sleep	Exit deep-sleep and standby
Alarm 0	ALRM0F	ALRM0IE and (ALRM0SECP=0 and RTCSECP=0)	write 1 in ALRM0FC	Y	Y ^(*)
Alarm 1	ALRM1F	ALRM1IE and (ALRM0SECP=0 and RTCSECP=0)	write 1 in ALRM0FC	Y	Y ^(*)
Wakeup	WTF	WTIE and (WUTSECP =0 and RTCSECP=0)	write 1 in WTFC	Y	Y ^(*)
Timestamp	TSF	TSIE and (TSSECP =0 and RTCSECP=0)	write 1 in TSFC	Y	Y ^(*)
Tamper 0	TP0F	TPIE and (TAMPSECP=0 and RTCSECP=0)	Write 1 in TP0FC	Y	Y ^(*)

NOTE: (*)Only active when RTC clock source is LXTAL or IRC32K.

16.4. Register definition

RTC secure access base address: 0x5000 2800

RTC non-secure access base address: 0x4000 2800

16.4.1. Time register (RTC_TIME)

Address offset: 0x00

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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								PM	HRT[1:0]		HRU[3:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]		MNU[3:0]			Reserved	SCT[2:0]		SCU[3:0]				rw		
	rw		rw				rw		rw				rw		

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PM	AM/PM mark 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15:	Reserved	Must be kept at reset value.
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value.
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

16.4.2. Date register (RTC_DATE)

Address offset: 0x04

System reset value: 0x0000 2101 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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								YRT[3:0]				YRU[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]				MONT		MONU[3:0]				Reserved		DAYT[1:0]		DAYU[3:0]	
rw		rw			rw							rw		rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	YRT	Year tens in BCD code
19:16	YRU[3:0]	Year units in BCD code
15:13	DOW[2:0]	Days of the week
		0x0: Reserved
		0x1: Monday
		...
		0x7: Sunday
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value.
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

16.4.3. Control register (RTC_CTL)

Address offset: 0x08

System reset: not affected

Backup domain reset value: 0x0000 0000

This register is writing protected, the register can be protected globally or individually per bit can be configured to prevent non-secure access or non-privileged access.

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	
OUT2EN	Reserved								COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
rw									rw		rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TSIE	WTIE	ALRM1IE	ALRMOIE	TSEN	WTEN	ALRM1E_N	ALRMOE_N	CCEN	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]			

rw rw

Bits	Fields	Descriptions
31	OUT2EN	RTC_OUT pin select 0: RTC_OUT output to PC15 1: RTC_OUT output to PA3 or PA8
30:24	Reserved	Must be kept at reset value.
23	COEN	Calibration output enable 0: Disable calibration output 1: Enable calibration output
22:21	OS[1:0]	Output selection This bit is used for selecting flag source to output 0x0: Disable output RTC_ALARM 0x1: Enable alarm0 flag output 0x2: Enable alarm1 flag output 0x3: Enable wakeup flag output
20	OPOL	Output polarity This bit is used to invert output RTC_ALARM 0: Disable invert output RTC_ALARM 1: Enable invert output RTC_ALARM
19	COS	Calibration output selection Valid only when COEN=1 and prescalers are at default values 0: Calibration output is 512 Hz 1: Calibration output is 1Hz
18	DSM	Daylight saving mark This bit is flexible used by software. Often can be used to recording the daylight saving hour adjustment.
17	S1H	Subtract 1 hour(winter time change) One hour will be subtracted from current time if it is not 0 0: No effect 1: 1 hour will be subtracted at next second change time.
16	A1H	Add 1 hour(summer time change) One hour will be added from current time 0: No effect 1: 1 hour will be added at next second change time
15	TSIE	Time-stamp interrupt enable 0: Disable time-stamp interrupt 1: Enable time-stamp interrupt

14	WTIE	Auto-wakeup timer interrupt enable 0: Disable auto-wakeup timer interrupt 1: Enable auto-wakeup timer interrupt
13	ALRM1IE	RTC alarm-1 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
12	ALRM0IE	RTC alarm-0 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
11	TSEN	Time-stamp function enable 0: Disable time-stamp function 1: Enable time-stamp function
10	WTEN	Auto-wakeup timer function enable 0: Disable function 1: Enable function
9	ALRM1EN	Alarm-1 function enable 0: Disable alarm function 1: Enable alarm function
8	ALRM0EN	Alarm-0 function enable 0: Disable alarm function 1: Enable alarm function
7	CCEN	Coarse calibration function enable 0: Disable function 1: Enable function Note: FACTOR_A must be greater than 6 before enabled and can only be written in initialization state.
6	CS	Clock System 0: 24-hour format 1: 12-hour format Note: Can only be written in initialization state
5	BPSHAD	Shadow registers bypass control 0: Reading calendar from shadow registers 1: Reading calendar from current real-time calendar Note: If frequency of APB1 clock is less than seven times the frequency of RTCCLK, this bit must set to 1.
4	REFEN	Reference clock detection function enable 0: Disable reference clock detection function 1: Enable reference clock detection function

Note: Can only be written in initialization state and FACTOR_S must be 0x00FF

3	TSEG	Valid event edge of time-stamp 0: rising edge is valid event edge for time-stamp event 1: falling edge is valid event edge for time-stamp event
2:0	WTCS[2:0]	Auto-wakeup timer clock selection 0x0:RTC Clock divided by 16 0x1:RTC Clock divided by 8 0x2:RTC Clock divided by 4 0x3:RTC Clock divided by 2 0x4:0x5: ck_spre (default 1Hz) clock 0x6:0x7: ck_spre (default 1Hz) clock and 2^{16} is added to wake-up counter.

16.4.4. Initialization control and status register (RTC_ICSR)

Address offset: 0x0C

System reset: Only INITM, INITF and RSYNF bits are set to 0. Others are not affected

Backup domain reset value: 0x0000 0007

This register is writing protected, the register can be protected globally or individually per bit can be configured to prevent non-secure access or non-privileged access.

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															SCPF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															ALRM1WF ALRM0WF
								rw	r	rc_w0	r	r	r	r	r

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	SCPF	Smooth calibration pending flag Set to 1 by hardware when software writes to RTC_HRFC without entering initialization mode and set to 0 by hardware when smooth calibration configuration is taken into account.
15:8	Reserved	Must be kept at reset value.
7	INITM	Enter initialization mode 0: Free running mode 1: Enter initialization mode for setting calendar time/date and prescaler. Counter will stop under this mode.
6	INITF	Initialization state flag Set to 1 by hardware and calendar register and prescaler can be programmed in

		this state.
		0: Calendar registers and prescaler register cannot be changed 1: Calendar registers and prescaler register can be changed
5	RSYNF	<p>Register synchronization flag</p> <p>Set to 1 by hardware every 2 RTCCLK which will copy current calendar time/date into shadow register. Initialization mode (INITM), shift operation pending flag (SOPF) or bypass mode (BPSHAD) will clear this bit. This bit is also can be cleared by software writing 0.</p> <p>0: Shadow register are not yet synchronized 1: Shadow register are synchronized</p>
4	YCM	<p>Year configuration mark</p> <p>Set by hardware if the year field of calendar date register is not the default value 0.</p> <p>0: Calendar has not been initialized 1: Calendar has been initialized</p>
3	SOPF	<p>Shift function operation pending flag</p> <p>0: No shift operation is pending 1: Shift function operation is pending</p>
2	WTWF	<p>Wakeup timer write enable flag</p> <p>0: Wakeup timer update is not allowed 1: Wakeup timer update is allowed</p>
1	ALRM1WF	<p>Alarm 1 configuration can be write flag</p> <p>Set by hardware if alarm register can be wrote after ALRM1EN bit has reset.</p> <p>0: Alarm registers programming is not allowed 1: Alarm registers programming is allowed</p>
0	ALRM0WF	<p>Alarm 0 configuration can be write flag</p> <p>Set by hardware if alarm register can be wrote after ALRM0EN bit has reset.</p> <p>0: Alarm registers programming is not allowed. 1: Alarm registers programming is allowed.</p>

16.4.5. Prescaler register (RTC_PSC)

Address offset: 0x10

System reset: not effected

Backup domain reset value: 0x007F 00FF

This register is write protected and can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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								FACTOR_A[6:0]							

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22:16	FACTOR_A[6:0]	Asynchronous prescaler factor $\text{ck_apre frequency} = \text{RTCCLK frequency}/(\text{FACTOR_A}+1)$
15	Reserved	Must be kept at reset value.
14:0	FACTOR_S[14:0]	Synchronous prescaler factor $\text{ck_spre frequency} = \text{ck_apre frequency}/(\text{FACTOR_S}+1)$

16.4.6. Wakeup timer register (RTC_WUT)

Address offset: 0x14

System reset: not effected

Backup domain reset value: 0x0000 FFFF

This register can be write-protected to prevent non-secure access or non-privileged access.

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
WTRV[15:0]															
FW															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTRV[15:0]	<p>Auto-wakeup timer reloads value. Every (WTRV[15:0]+1) ck_wut period the WTF bit is set after WTEN=1. The ck_wut is selected by WTCS[2:0] bits.</p> <p>Note: This configure case is forbidden: WTRV=0x0000 with WTCS[2:0]=0b011.</p> <p>This register can be written only when WTWF=1.</p>

16.4.7. Coarse calibration register (RTC_COSC)

Address offset: 0x18

System reset: not effect

Backup domain reset value: 0x0000 0000

This register is write protected and can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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								COSD	Reserved		COSS[4:0]				

rw

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	COSD	Coarse Calibration direction 0: Increase calendar update frequency 1: Decrease calendar update frequency
6:5	Reserved	Must be kept at reset value.
4:0	COSS[4:0]	Coarse Calibration step When COSD=0: 0x00:+0 PPM 0x01:+4 PPM(approximate value) 0x02:+8 PPM(approximate value) 0x1F:+126 PPM(approximate value) When COSD=1: 0x00:-0 PPM 0x01:-2 PPM(approximate value) 0x02:-4 PPM(approximate value) 0x1F:-63 PPM(approximate value)

16.4.8. Alarm 0 time and date register (RTC_ALRM0TD)

Address offset: 0x1C

System reset: not effect

Backup domain reset value: 0x0000 0000

This register can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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
MSKD	DOWS	DAYT[1:0]			DAYU[3:0]			MSKH	PM	HRT[1:0]			HRU[3:0]		
rw	rw	rw			rw			rw	rw	rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSKM	MNT[2:0]	MNU[3:0]	MSKS	SCT[2:0]	SCU[3:0]
rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	MSKD	Alarm date mask bit 0: Not mask date/day field 1: Mask date/day field
30	DOWS	Day of the week selected 0: DAYU[3:0] indicates the date units 1: DAYU[3:0] indicates the week day and DAYT[1:0] has no means.
29:28	DAYT[1:0]	Date tens in BCD code
27:24	DAYU[3:0]	Date units or week day in BCD code
23	MSKH	Alarm hour mask bit 0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes mask bit 0: Not mask minutes field 1: Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0: Not mask second field 1: Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

16.4.9. Alarm 1 time and date register (RTC_ALRM1TD)

Address offset: 0x20

System reset: not effect

Backup domain reset value: 0x0000 0000

This register can only be written in initialization state

This register can be write-protected to prevent non-secure access or non-privileged access

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
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]			MSKH	PM	HRT[1:0]		HRU[3:0]				
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
MSKM	MNT[2:0]		MNU[3:0]			MSKS	SCT[2:0]		SCU[3:0]						
rw	rw		rw			rw	rw	rw		rw		rw			

Bits	Fields	Descriptions
31	MSKD	Alarm date mask bit 0: Not mask date/day field 1: Mask date/day field
30	DOWS	Day of the week selected 0: DAYU[3:0] indicates the date units 1: DAYU[3:0] indicates the week day and DAYT[3:0] has no means.
29:28	DAYT[1:0]	Day tens in BCD code
27:24	DAYU[3:0]	Day units or week day in BCD code
23	MSKH	Alarm hour mask bit 0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes mask bit 0: Not mask minutes field 1: Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0: Not mask second field 1: Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

16.4.10. Write protection key register (RTC_WPK)

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	WPK[7:0]	Key for write protection

16.4.11. Sub second register (RTC_SS)

Address offset: 0x28

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

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
SSC[15:0]															

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler. Second fraction value is calculated by the below formula: $\text{Second fraction} = (\text{FACTOR_S} - \text{SSC}) / (\text{FACTOR_S} + 1)$

16.4.12. Shift function control register (RTC_SHIFTCTL)

Address offset: 0x2C

System reset: not effect

Backup Reset value: 0x0000 0000

This register is writing protected and can only be wrote when SOPF=0

This register can be write-protected to prevent non-secure access or non-privileged access

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
A1S	Reserved														
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SFS[14:0]														
w															

Bits	Fields	Descriptions
31	A1S	One second add 0: Not add 1 second 1: Add 1 second to the clock/calendar. This bit is jointly used with SFS field to add a fraction of a second to the clock.
30:15	Reserved	Must be kept at reset value.
14:0	SFS[14:0]	Subtract a fraction of a second The value of this bit will add to the counter of synchronous prescaler. When only using SFS, the clock will delay because the synchronous prescaler is a down counter: Delay (seconds) = SFS / (FACTOR_S + 1) When jointly using A1S and SFS, the clock will advance: Advance (seconds) = (1 - (SFS / (FACTOR_S + 1)))

Note: Writing to this register will cause RSYNF bit to be cleared.

16.4.13. Time of time stamp register (RTC_TTS)

Address offset: 0x30

Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar time when TSF is set to 1.

Reset TSF bit will also clear this register.

This register can be write-protected to prevent non-secure access or non-privileged access

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					PM	HRT[1:0]			HRU[3:0]						
						r		r					r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]			MNU[3:0]			Reserved	SCT[2:0]			SCU[3:0]				
	r			r				r				r			

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.

22	PM	AM/PM mark 0:AM or 24-hour format 1:PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	Reserved	Must be kept at reset value.
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value.
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

16.4.14. Date of time stamp register (RTC_DTS)

Address offset: 0x34

Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar date when TSF is set to 1.

This register can be write-protected to prevent non-secure access or non-privileged access

Reset TSF bit will also clear this register.

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
DOW[2:0]	MONT	MONU[3:0]	Reserved	DAYT[1:0]	DAYU[3:0]										
r	r	r		r	r										

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:13	DOW[2:0]	Days of the week
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value.
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

16.4.15. Sub second of time stamp register (RTC_SSTS)

Address offset: 0x38

Backup domain reset: 0x0000 0000

System reset: no effect

This register will record the calendar date when TSF is set to 1.

Reset TSF bit will also clear this register.

This register can be write-protected to prevent non-secure access or non-privileged access

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
SSC[15:0]															

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF is set to 1.

16.4.16. High resolution frequency compensation register (RTC_HRFC)

Address offset: 0x3C

Backup domain reset: 0x0000 0000

System Reset: no effect

This register is write protected.

This register can be write-protected to prevent non-secure access or non-privileged access

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
FREQI	CWND8	CWND16	Reserved				CMSK[8:0]								rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FREQI	Increase RTC frequency by 488.5PPM 0: No effect 1: One RTCCLK pulse is inserted every 2^{11} pulses. This bit should be used in conjunction with CMSK bit. If the input clock frequency is

32.768KHz, the number of RTCCLK pulses added during 32s calibration window is $(512 * \text{FREQI}) - \text{CMSP}$

14	CWND8	Frequency compensation window 8 second selected 0: No effect 1: Calibration window is 8 second Note: When CWND8=1, CMSK[1:0] are stuck at "00".
13	CWND16	Frequency compensation window 16 second selected 0: No effect 1: Calibration window is 16 second Note: When CWND16=1, CMSK[0] are stuck at "0".
12:9	Reserved	Must be kept at reset value.
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537 PPM.

16.4.17. Tamper register (RTC_TAMP)

Address offset: 0x40

Backup domain reset: 0x0000 0000

System reset: no effect

This register can be write-protected to prevent non-secure access or non-privileged access

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
BKERAS E											TP1NOER	TP0NOER	AOT		Reserved
rw											rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]	FLT[1:0]		FREQ[2:0]	TPTS	Reserved	TP1EG	TP1EN	TP1IE	TP0EG	TP0EN				
rw	rw	rw		rw	rw		rw	rw	rw	rw	rw				

Bits	Fields	Descriptions
31	BKERASE	Backup registers erase Writing '1' to this bit reset the backup registers. Writing 0 has no effect. This bit is always read as 0.
30:21	Reserved	Must be kept at reset value.
20	TP1NOER	Tamper 1 no erase 0: Tamper 1 event erases the backup registers. 1: Tamper 1 event does not erase the backup registers
19	TP0NOER	Tamper 0 no erase 0: Tamper 0 event erases the backup registers.

		1: Tamper 0 event does not erase the backup registers
18	AOT	<p>RTC_ALARM Output Type</p> <p>0: Open-drain output type</p> <p>1: Push-pull output type</p>
17:16	Reserved	Must be kept at reset value.
15	DISPU	<p>RTC_TAMPx pull up disable bit</p> <p>0: Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin</p> <p>1: Disable pre-charge duration</p>
14:13	PRCH[1:0]	<p>Pre-charge duration time of RTC_TAMPx</p> <p>This setting determines the pre-charge time before each sampling.</p> <p>0x0: 1 RTC clock</p> <p>0x1: 2 RTC clock</p> <p>0x2: 4 RTC clock</p> <p>0x3: 8 RTC clock</p>
12:11	FLT[1:0]	<p>RTC_TAMPx filter count setting</p> <p>This bit determines the tamper sampling type and the number of consecutive sample.</p> <p>0x0: Detecting tamper event using edge mode. Pre-charge duration is disabled automatically</p> <p>0x1: Detecting tamper event using level mode. 2 consecutive valid level samples will make an effective tamper event</p> <p>0x2: Detecting tamper event using level mode. 4 consecutive valid level samples will make an effective tamper event</p> <p>0x3: Detecting tamper event using level mode. 8 consecutive valid level samples will make an effective tamper event</p>
10:8	FREQ[2:0]	<p>Sampling frequency of tamper event detection</p> <p>0x0: Sample once every 32768 RTCCLK(1Hz if RTCCLK=32.768KHz)</p> <p>0x1: Sample once every 16384 RTCCLK(2Hz if RTCCLK=32.768KHz)</p> <p>0x2: Sample once every 8192 RTCCLK(4Hz if RTCCLK=32.768KHz)</p> <p>0x3: Sample once every 4096 RTCCLK(8Hz if RTCCLK=32.768KHz)</p> <p>0x4: Sample once every 2048 RTCCLK(16Hz if RTCCLK=32.768KHz)</p> <p>0x5: Sample once every 1024 RTCCLK(32Hz if RTCCLK=32.768KHz)</p> <p>0x6: Sample once every 512 RTCCLK(64Hz if RTCCLK=32.768KHz)</p> <p>0x7: Sample once every 256 RTCCLK(128Hz if RTCCLK=32.768KHz)</p>
7	TPTS	<p>Make tamper function used for timestamp function</p> <p>0: No effect</p> <p>1: TSF is set when tamper event detected even TSEN=0</p>
6:5	Reserved	Must be kept at reset value.

4	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
3	TP1EN	Tamper 1 detection enable 0: Disable tamper 1 detection function 1: Enable tamper 1 detection function
2	TP1IE	Tamper detection interrupt enable 0: Disable tamper interrupt 1: Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0: Disable tamper 0 detection function 1: Enable tamper 0 detection function

Note: It's strongly recommended that reset the TPxEN before change the tamper configuration.

16.4.18. Alarm 0 sub second register (RTC_ALRM0SS)

Address offset: 0x44

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM0EN=0 or INITM=1

This register can be write-protected to prevent non-secure access or non-privileged access

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

Bits	Fields	Descriptions
------	--------	--------------

31:28	Reserved	Must be kept at reset value.
27:24	MSKSSC[3:0]	<p>Mask control bit of SSC</p> <p>0x0: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched.</p> <p>0x1: SSC[0] is to be compared and all others are ignored</p> <p>0x2: SSC[1:0] is to be compared and all others are ignored</p> <p>0x3: SSC[2:0] is to be compared and all others are ignored</p> <p>0x4: SSC[3:0] is to be compared and all others are ignored</p> <p>0x5: SSC[4:0] is to be compared and all others are ignored</p> <p>0x6: SSC[5:0] is to be compared and all others are ignored</p> <p>0x7: SSC[6:0] is to be compared and all others are ignored</p> <p>0x8: SSC[7:0] is to be compared and all others are ignored</p> <p>0x9: SSC[8:0] is to be compared and all others are ignored</p> <p>0xA: SSC[9:0] is to be compared and all others are ignored</p> <p>0xB: SSC[10:0] is to be compared and all others are ignored</p> <p>0xC: SSC[11:0] is to be compared and all others are ignored</p> <p>0xD: SSC[12:0] is to be compared and all others are ignored</p> <p>0xE: SSC[13:0] is to be compared and all others are ignored</p> <p>0xF: SSC[14:0] is to be compared and all others are ignored</p> <p>Note: The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.</p>
23:15	Reserved	Must be kept at reset value.
14:0	SSC[14:0]	<p>Alarm sub second value</p> <p>This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.</p>

16.4.19. Alarm 1 sub second register (RTC_ALRM1SS)

Address offset: 0x48

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM1EN=0 or INITM=1

This register can be write-protected to prevent non-secure access or non-privileged access

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

Bits	Fields	Descriptions
------	--------	--------------

31:28	Reserved	Must be kept at reset value.
27:24	MSKSSC[3:0]	<p>Mask control bit of SSC</p> <p>0x0: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched.</p> <p>0x1: SSC[0] is to be compared and all others are ignored</p> <p>0x2: SSC[1:0] is to be compared and all others are ignored</p> <p>0x3: SSC[2:0] is to be compared and all others are ignored</p> <p>0x4: SSC[3:0] is to be compared and all others are ignored</p> <p>0x5: SSC[4:0] is to be compared and all others are ignored</p> <p>0x6: SSC[5:0] is to be compared and all others are ignored</p> <p>0x7: SSC[6:0] is to be compared and all others are ignored</p> <p>0x8: SSC[7:0] is to be compared and all others are ignored</p> <p>0x9: SSC[8:0] is to be compared and all others are ignored</p> <p>0xA: SSC[9:0] is to be compared and all others are ignored</p> <p>0xB: SSC[10:0] is to be compared and all others are ignored</p> <p>0xC: SSC[11:0] is to be compared and all others are ignored</p> <p>0xD: SSC[12:0] is to be compared and all others are ignored</p> <p>0xE: SSC[13:0] is to be compared and all others are ignored</p> <p>0xF: SSC[14:0] is to be compared and all others are ignored</p> <p>Note: The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.</p>
23:15	Reserved	Must be kept at reset value.
14:0	SSC[14:0]	<p>Alarm sub second value</p> <p>This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.</p>

16.4.20. Privilege protection mode control register (RTC_PPM_CTL)

Address offset: 0x50

Backup domain reset: 0x0000 0000

System reset: no effect

Register access reference [RTC privilege protection](#).

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		
BKPWPRIP	BKPRWP RIP	Reserved															
rw	rw																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RTCPRIIP	INTPRIIP	CALCPRI P	TAMPPRI P	Reserved													
rw	rw	rw	rw														

Bits	Fields	Descriptions
31	BKPWPRIP	Backup registers zone B privilege protection

When set, backup registers zone B can be written only when the APB access is in privileged mode, otherwise both APB access is in privileged and non-privileged mode can write backup domain registers.

30	BKPRWPRIP	Backup registers zone A privilege protection When set, backup registers zone A can be read and written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can read and written backup domain registers.
29:16	Reserved	Must be kept at reset value.
15	RTCP RIP	RTC privilege protection When set, all RTC registers can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written all RTC registers (except the registers protected by other privilege protection bits).
14	INITPRIP	Initialization privilege protection When set, RTC_TIME, RTC_DATE and RTC_PSC registers can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written above registers. Refer to <u>Table 16-4. RTC privileged protected mode configuration summary</u> .
13	CALCPRIP	Shift register, Delight saving, calibration and reference clock privilege protection When set, RTC_SHIFTCTL, RTC_COSC, RTC_CTL registers can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written the above registers. Refer to <u>Table 16-4. RTC privileged protected mode configuration summary</u> .
12	TAMPPRIP	Tamper privilege protection (excluding backup registers) When set, tamper configuration and interrupt can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written the above configuration.
11:4	Reserved	Must be kept at reset value.
3	TSP RIP	Timestamp privilege protection When set, timestamp configuration and interrupt clear can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written the above configuration.
2	WUTPRIP	Wakeup timer privilege protection When set, wakeup timer configuration and interrupt clear can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or non-privileged mode can be written the above configuration.
1	ALRM1PRIP	Alarm 1 privilege protection When set, Alarm 1 configuration and interrupt clear can be written only when the APB access is in privileged mode, otherwise whether APB access is in privileged or

non-privileged mode can be written the above configuration.

0	ALRM0PRIP	Alarm 0 privilege protection When set, Alarm 0 configuration and interrupt clear can be written only when the APB access is in privileged mode, otherwise APB access is in privileged or non-privileged mode can be written the above configuration.
---	-----------	---

16.4.21. Secure protection mode control register (RTC_SPM_CTL)

Address offset: 0x54

Backup domain reset: 0x0000 E00F

System reset: no effect

Register access reference [RTC security protection](#).

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
BKPWSECP[7:0]								BKPRWSECP[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCSEC P	INITSEC P	CALCSEC CP	TAMPSEC CP	Reserved								TSSECP P	WUTSEC P	ALRM1S ECP	ALRM0S ECP
rw	rw	rw	rw									rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	BKPWSECP[7:0]	Backup registers write protection offset If TZEN=1: backup registers from RTC_BKP t ($t = BKPRWSECP$, from 0 to 128) to RTC_BKP m ($m = BKPWSECP - 1$, from 0 to 128, $BKPWSECP \geq BKPRWSECP$) can be written only when the APB is in secure mode. They can be read in secure or non-secure mode. This zone is the protection zone B. If TZEN=0: the protection zone B can be read and written with non-secure access. Backup registers from RTC_BKP m ($m = BKPWSECP$, from 0 to 127) can be read or written when the APB is in secure or in non-secure mode. This zone is the protection zone C. If $BKPWSECP = 0$ or if $BKPWSECP \leq BKPRWSECP$: none of the backup registers have a secure write access. In these configurations the behavior is equivalent to $BKPWSECP = BKPRWSECP$. If BKPRWPRIP is set, BKPRWSECP[7:0] and BKPWSECP[7:0] can be written only in privileged mode.
23:16	BKPRWSECP[7:0]	Backup registers read/write protection offset If TZEN=1: backup registers from RTC_BKP0 to RTC_BKP t ($t = BKPRWSECP - 1$, from 0 to 128) can be read and written only when the APB is in secure mode. This is the protection zone A. If TZEN=0: the protection zone A can be read and written with non-secure access. If BKPRWSECP = 0 none of the backup registers have a secure read/write access. If BKPRWPRIP is set, BKPRWSECP[7:0] can be written only in privileged mode.
15	RTCSECP	RTC global protection

		When set, all RTC registers can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written all RTC registers (except the registers protected by other secure protection bits).
14	INITSECP	Initialization protection When set, RTC_TIME, RTC_DATE and RTC_PSC registers can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written above registers. Refer to <u>Table 16-2. RTC secure mode configuration summary</u>
13	CALSECP	Shift register, daylight saving, calibration and reference clock protection When set, RTC_SHIFTCTL, RTC_COSC, RTC_CTL registers can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above registers. Refer to <u>Table 16-2. RTC secure mode configuration summary</u>
12	TAMPSECP	Tamper protection (excluding backup registers) When set, tamper configuration and interrupt can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above configuration.
11:4	Reserved	Must be kept at reset value.
3	TSSECP	Timestamp protection When set, timestamp configuration and interrupt clear can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above configuration.
2	WUTSECP	Wakeup timer protection When set, wakeup timer configuration and interrupt clear can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above configuration.
1	ALRM1SECP	Alarm 1 protection When set, Alarm 1 configuration and interrupt clear can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above configuration.
0	ALRM0SECP	Alarm 0 protection When set, Alarm 0 configuration and interrupt clear can be written only when the APB access is in secured mode, otherwise whether APB access is in secured or non-secured mode can be written the above configuration.

16.4.22. Status register (RTC_STAT)

Address offset: 0x58

Backup domain reset: 0x0000 0000

System reset: no effect

This register can be protected globally or individually per bit can be configured to prevent non-secure access or non-privileged access..

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														TP1F	TP0F			
														r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reserved														TSOVRF	TSF	WTF	ALRM1F	ALRM0F
														r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TP1F	RTC_TAMP1 detected flag Set to 1 by hardware when tamper detection is found on tamper1 input pin.
16	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin.
15:5	Reserved	Must be kept at reset value.
4	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before.
3	TSF	Time-stamp flag Set by hardware when time-stamp event is detected.
2	WTF	Wakeup timer flag Set by hardware when wakeup timer decreased to 0. This flag must be cleared at least 1.5 RTC Clock periods before WTF is set to 1 again.
1	ALRM1F	Alarm-1 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 1 setting value.
0	ALRM0F	Alarm-0 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value.

Note: The bits of this register are cleared 2 APB clock cycles after setting their corresponding clear bit in the RTC_SCR register.

16.4.23. Non-secure masked interrupt status register (RTC_NSMI_STAT)

Address offset: 0x5C

Backup domain reset: 0x0000 0000

System reset: no effect

This register can be protected globally or individually per bit can be configured to prevent non-privileged access.

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														TP1NSMF	TP0NSMF			
														r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reserved														TSOVRNSMF	TSNSMF	WTNSMF	ALRM1NSMF	ALRM0NSMF
														r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TP1NSMF	RTC_TAMP1 non-secure interrupt masked flag Set to 1 by hardware when tamper detection is found on tamper1 input pin.
16	TP0NSMF	RTC_TAMP0 non-secure interrupt masked flag Set to 1 by hardware when tamper detection is found on tamper0 input pin.
15:5	Reserved	Must be kept at reset value.
4	TSOVRNSMF	Time-stamp overflow non-secure masked flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before.
3	TSNSMF	Time-stamp flag Set by hardware when time-stamp event is detected.
2	WTNSMF	Wakeup timer non-secure masked flag Set by hardware when wakeup timer decreased to 0. This flag must be cleared at least 1.5 RTC Clock periods before WTF is set to 1 again.
1	ALRM1NSMF	Alarm-1 occurs non-secure masked flag Set to 1 by hardware when current time/date matches the time/date of alarm 1 setting value.
0	ALRM0NSMF	Alarm-0 occurs non-secure masked flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value.

16.4.24. Secure masked interrupt status register (RTC_SMI_STAT)

Address offset: 0x60

Backup domain reset: 0x0000 0000

System reset: no effect

This register can be protected globally or individually per bit can be configured to prevent non-secure access or non-privileged access.

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															TP1SMF	TP0SMF			
															r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reserved															TSOVRS MF	TSSMF	WTSMF	ALRM1S MF	ALRM0S MF
															r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TP1SMF	RTC_TAMP1 secure interrupt masked flag Set to 1 by hardware when tamper detection is found on tamper1 input pin.
16	TP0SMF	RTC_TAMP0 secure interrupt masked flag Set to 1 by hardware when tamper detection is found on tamper0 input pin.
15:5	Reserved	Must be kept at reset value.
4	TSOVRSMF	Time-stamp overflow secure masked flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before.
3	TSSMF	Time-stamp secure masked flag Set by hardware when time-stamp event is detected.
2	WTSMF	Wakeup timer secure masked flag Set by hardware when wakeup timer decreased to 0. This flag must be cleared at least 1.5 RTC Clock periods before WTF is set to 1 again.
1	ALRM1SMF	Alarm-1 occurs secure masked flag Set to 1 by hardware when current time/date matches the time/date of alarm 1 setting value.
0	ALRM0SMF	Alarm-0 occurs secure masked flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value.

16.4.25. Status flag clear register (RTC_STATC)

Address offset: 0x64

Backup domain reset: 0x0000 0000

System reset: no effect

This register can be protected globally or individually per bit can be configured to prevent non-secure access or non-privileged access.

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														TP1FC	TP0FC			
														W	W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reserved														TSOVRF C	TSFC	WTFC	ALRM1F C	ALRM0F C
														W	W	W	W	W

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TP1FC	TAMP1 detection flag clear Writing 1 in this bit clears the TP1F bit in the RTC_STAT register.
16	TP0FC	TAMP0 detection flag clear Writing 1 in this bit clears the TP0F bit in the RTC_STAT register.
15:5	Reserved	Must be kept at reset value.
4	TSOVRFC	Timestamp overflow flag clear. Writing 1 in this bit clears the TSOVRF bit in the RTC_STAT register. It is recommended to check and then clear TSOVRF only after clearing the TSF bit. Otherwise, an overflow might not be noticed if a timestamp event occurs immediately before the TSF bit is cleared.
3	TSFC	Timestamp flag clear. Writing 1 in this bit clears the TSF bit in the RTC_STAT register.
2	WTFC	Wakeup timer flag clear. Writing 1 in this bit clears the WTF bit in the RTC_STAT register.
1	ALRM1FC	Alarm 1 flag clear. Writing 1 in this bit clears the ALRM1F bit in the RTC_STAT register.
0	ALRM0FC	Alarm 0 flag clear. Writing 1 in this bit clears the ALRM0F bit in the RTC_STAT register.

16.4.26. Backup registers (RTC_BKPx) (x=0..19)

Address offset: 0x70~0xBC

Backup domain reset: 0x0000 0000

System reset: no effect

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
DATA[31:16]															
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]														rw	

Bits	Fields	Descriptions
31:0	DATA[31:0]	<p>Data</p> <p>These registers can be wrote or read by software. The content remains valid even in power saving mode because they can powered-on by V_{BAT}. Tamper detection flag TPxF assertion will reset these registers.</p>

17. Timer (TIMERx)

Table 17-1. Timers (TIMERx) are divided into four sorts

TIMER	TIMER0	TIMER1/2/3/4	TIMER15/16	TIMER5
TYPE	Advanced	General-L0	General-L4	Basic
Prescaler	16-bit	16-bit	16-bit	16-bit
Counter	16-bit	32-bit(TIMER1/2) 16-bit(TIMER3/4)	16-bit	16-bit
Count mode	UP,DOWN, Center-aligned	UP,DOWN, Center-aligned	UP ONLY	UP ONLY
Repetition	•	✗	•	✗
CH Capture/ Compare	4	4	1	0
Complementary & Dead-time	•	✗	•	✗
Break	•	✗	•	✗
Single Pulse	•	•	•	•
Quadrature Decoder	•	•	✗	✗
Slave Controller	•	•	✗	✗
Inter connection	• ⁽¹⁾	• ⁽²⁾	✗	✗
DMA	•	•	•	• ⁽³⁾
Debug Mode	•	•	•	•

(1) TIMER0 **ITI0: 0** **ITI1: TIMER1_TRGO** **ITI2: TIMER2_TRGO** **ITI3: 0**

(2) TIMER1 **ITI0: TIMER0_TRGO** **ITI1: 0** **ITI2: TIMER2_TRGO** **ITI3: 0**

ITI0: TIMER0_TRGO **ITI1: TIMER1_TRGO** **ITI2: 0** **ITI3: 0**

ITI0: TIMER0_TRGO **ITI1: TIMER1_TRGO** **ITI2: TIMER2_TRGO** **ITI3: 0**

ITI0: TIMER0_TRGO **ITI1: TIMER1_TRGO** **ITI2: TIMER14_TRGO** **ITI3: 0**

(3) Only update events will generate a DMA request. Note that TIMER5 do not have DMA configuration registers.

17.1. Advanced timer (TIMERx, x=0)

17.1.1. Overview

The advanced timer module (Timer0) is a four-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.

Timer and timer are completely independent with each other, but they may be synchronized to provide a larger timer with their counters incrementing in unison.

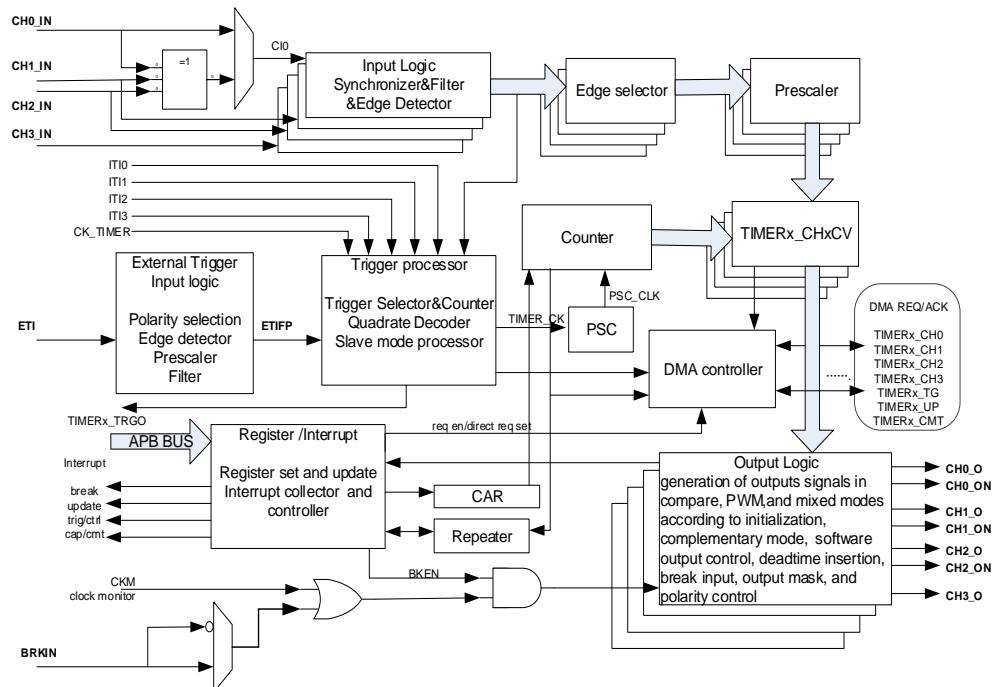
17.1.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bit.
- Source of counter clock is selectable:
internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: count up, count down, count up/down.
- Quadrature Decoder: used to track motion and determine both rotation direction and position.
- Hall sensor: for 3-phase motor control.
- Programmable prescaler: 16 bit. The factor can be changed on the go.
- Each channel is user-configurable:
input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request on: update, trigger event, compare/capture event, and break input.
- Daisy chaining of timer modules allows a single timer to initiate multiple timers.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer Master/Slave mode controller.

17.1.3. Block diagram

[Figure 17-1. Advanced timer block diagram](#) provides details of the internal configuration of the advanced timer.

Figure 17-1. Advanced timer block diagram



17.1.4. Function overview

Clock selection

The advanced timer has the capability of being clocked by either the TIMER_CK or an alternate clock source controlled by SMC (TIMERx_SMCFG bit [2:0]).

- SMC [2:0] == 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

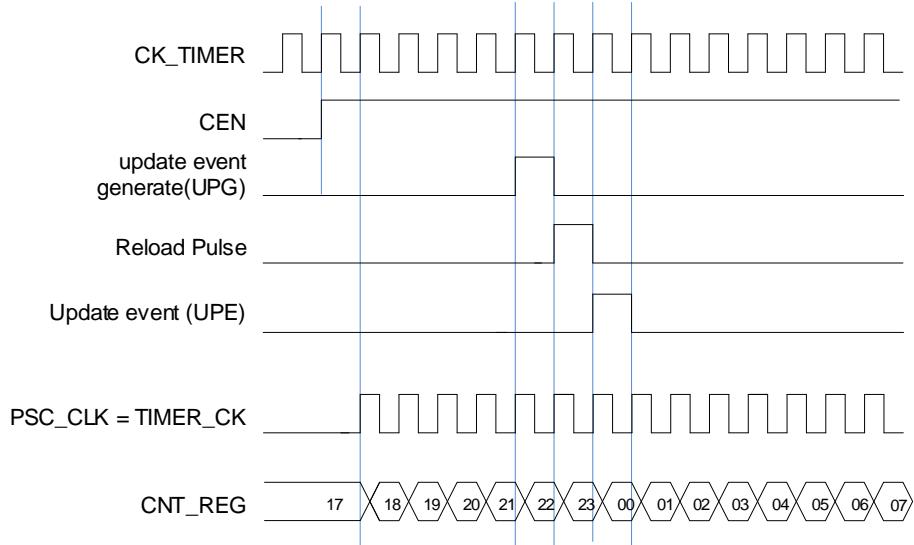
The default clock source is the CK_TIMER for driving the counter prescaler when the slave mode is disabled (SMC [2:0] == 3'b000). When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK, which drives counter's prescaler to count, is equal to CK_TIMER which is from RCU.

If the slave mode controller is enabled by setting SMC [2:0] in the TIMERx_SMCFG register to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock

sources selected by the TRGS [2:0] in the TIMERx_SMCFG register, details as follows. When the slave mode selection bits SMC [2:0] are set to 0x4, 0x5 or 0x6, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 17-2. Normal mode, internal clock divided by 1



- SMC[2:0]==3'b111 (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 TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting SMC [2:0] to 0x7 and the TRGS [2:0] to 0x4, 0x5 or 0x6.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting SMC [2:0] to 0x7 and the TRGS [2:0] to 0x0, 0x1, 0x2 or 0x3.

- SMC1==1'b1 (external clock mode 1). External input is selected as timer clock source (ETI)

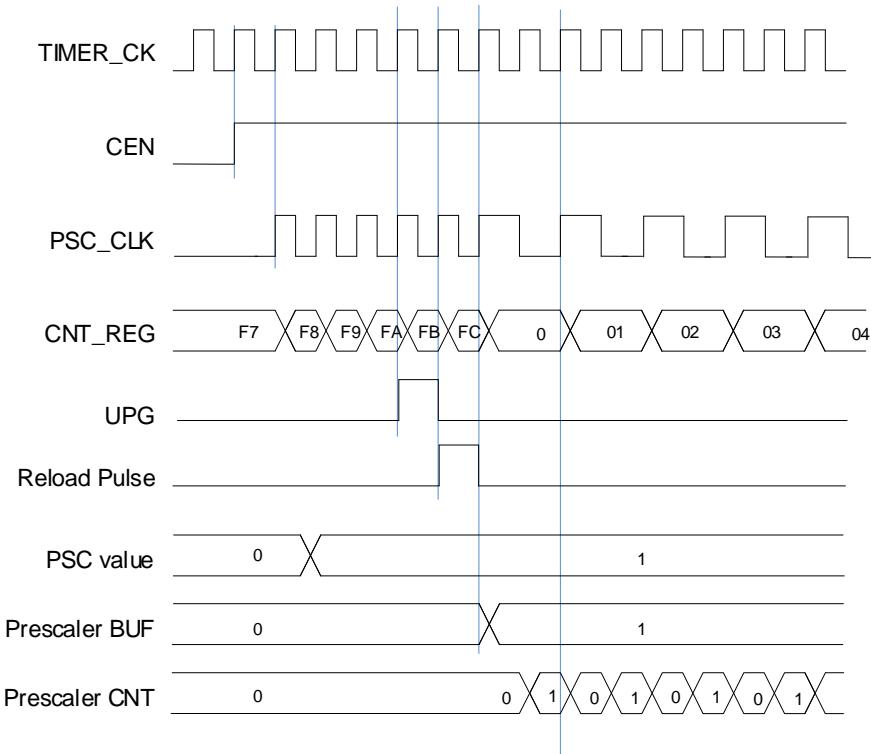
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is to set the SMC [2:0] to 0x7 and the TRGS [2:0] to 0x7 respectively. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as 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.

Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled by prescaler register (TIMERx_PSC) which can

be changed on the go but is taken into account at the next update event.

Figure 17-3. 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 `TIMERx_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 events will be generated after $(\text{TIMERx_CREP}+1)$ times of overflow. Otherwise the update event is generated each time when 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 generates an update event.

If set the `UPDIS` bit in `TIMERx_CTL0` register, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

[Figure 17-4. Timing chart of up counting mode, PSC=0/1](#) and [Figure 17-5. Timing chart of up counting mode, change `TIMERx_CAR` ongoing](#) show some examples of the counter behavior for different clock prescaler factor when `TIMERx_CAR`=0x63.

Figure 17-4. Timing chart of up counting mode, PSC=0/1

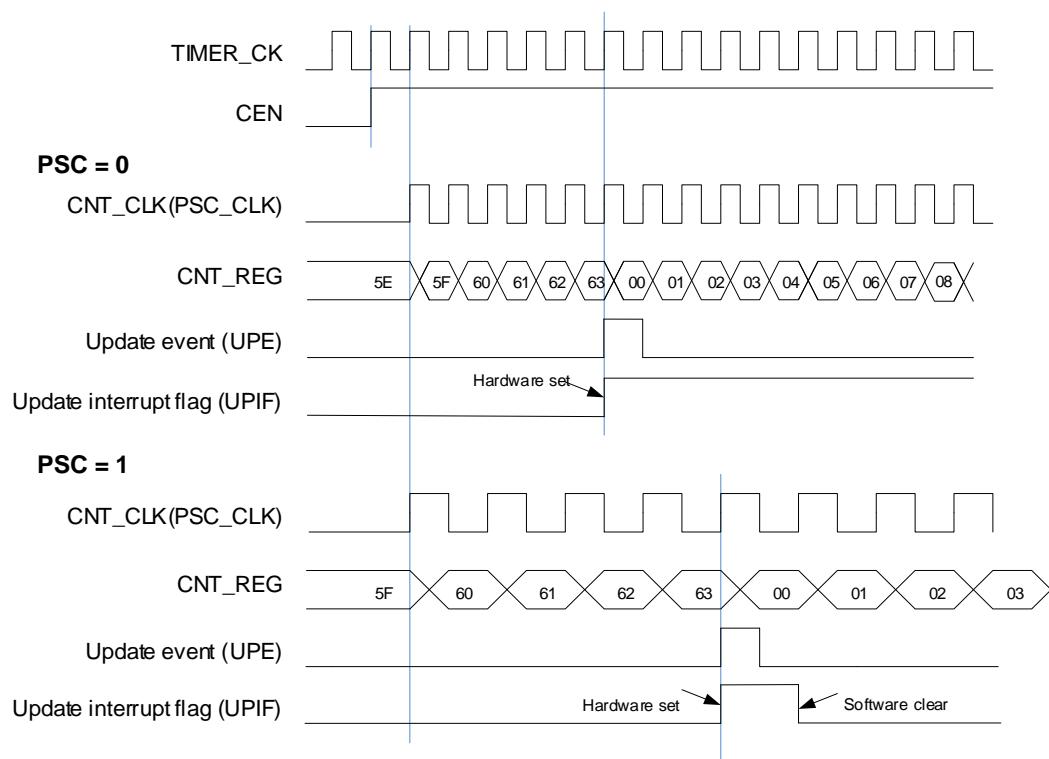
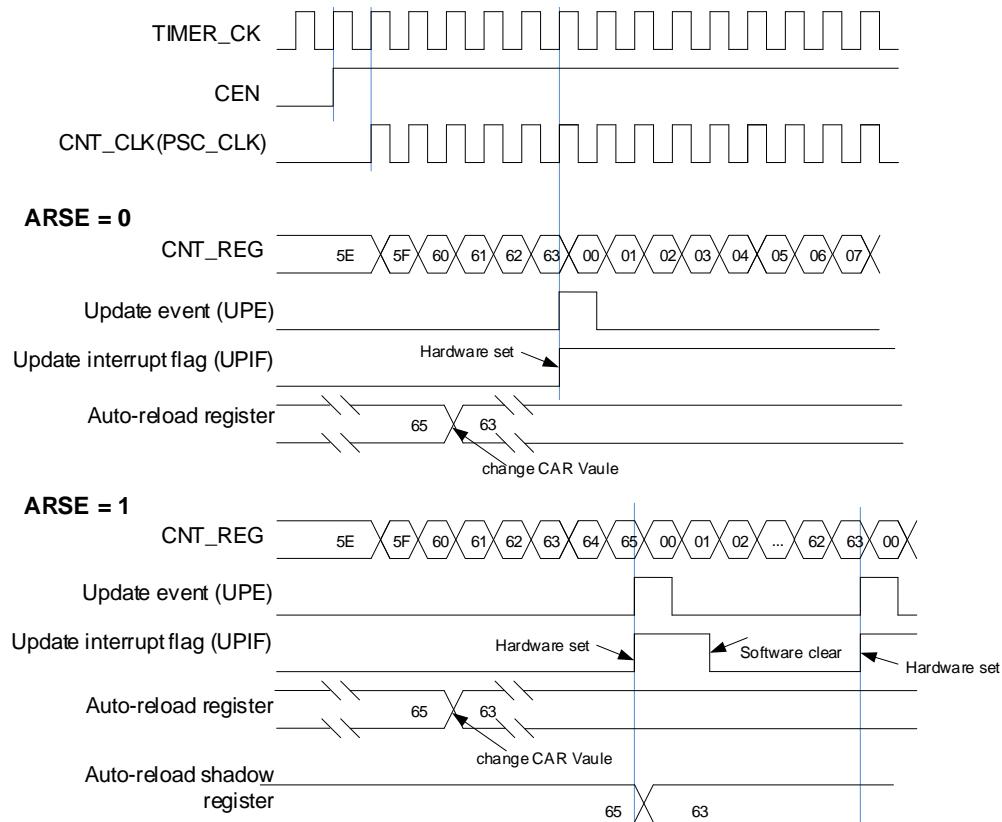


Figure 17-5. Timing chart of up counting mode, change TIMERx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter-reload value, which is defined in the TIMERx_CAR register, to 0 in a count-down direction. Once the counter reaches to 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 (TIMERx_CREP+1) times of underflow. Otherwise the update event is generated each time when 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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

[Figure 17-6. Timing chart of down counting mode, PSC=0/1](#) and [Figure 17-7. Timing chart of down counting mode, change TIMERx_CAR ongoing](#) show some examples of the counter behavior in different clock frequencies when TIMERx_CAR=0x63.

Figure 17-6. Timing chart of down counting mode, PSC=0/1

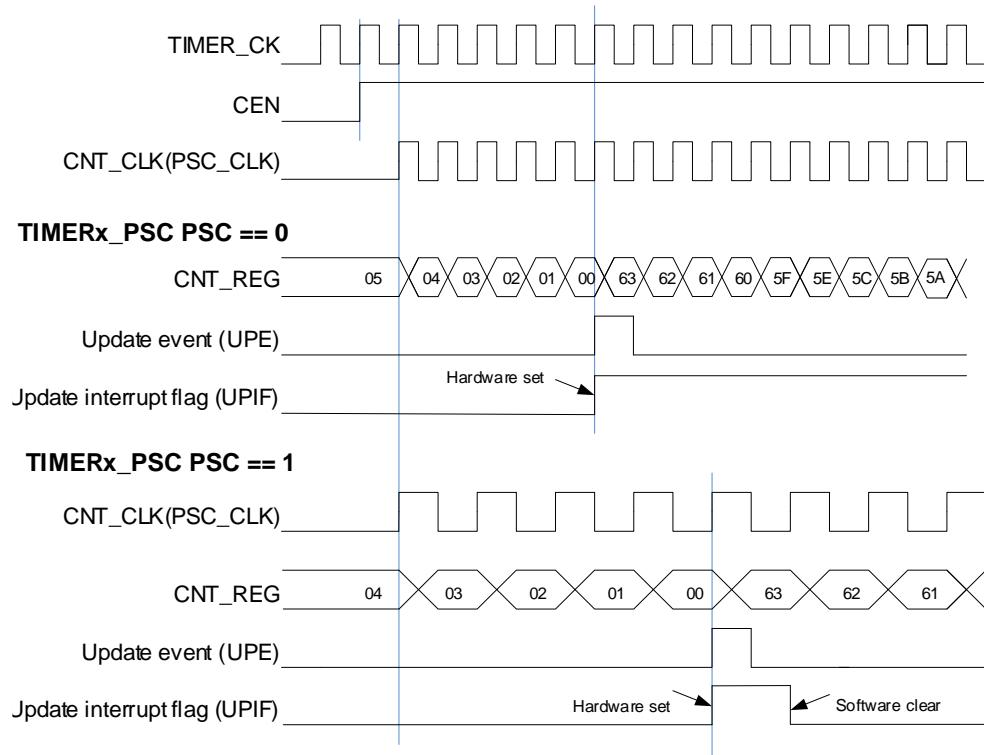
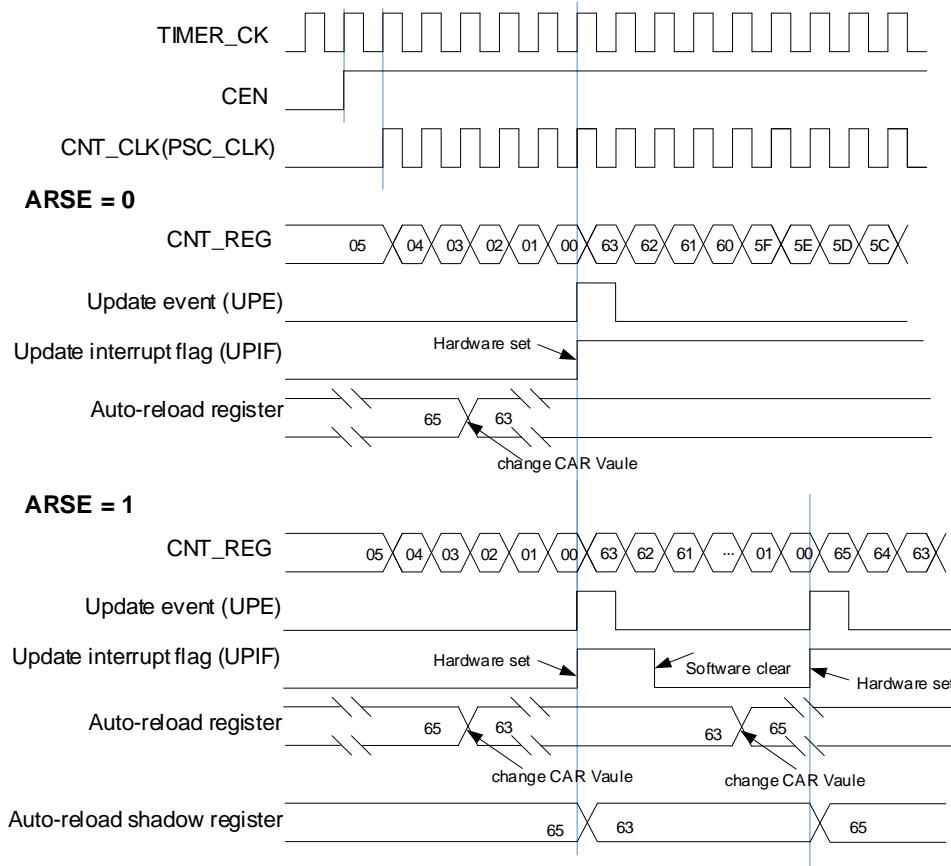


Figure 17-7. Timing chart of down counting mode, change TIMERx_CAR ongoing



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 the counter-reload value subtract 1 in the up-counting direction and generates an underflow event when the counter counts to 1 in the down-counting direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned 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 generates an update event irrespective of whether the counter is counting up or down in the center-align counting mode.

The UPIF bit in the TIMERx_INTF register can 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 17-8. Timing chart of center-aligned counting mode.](#)

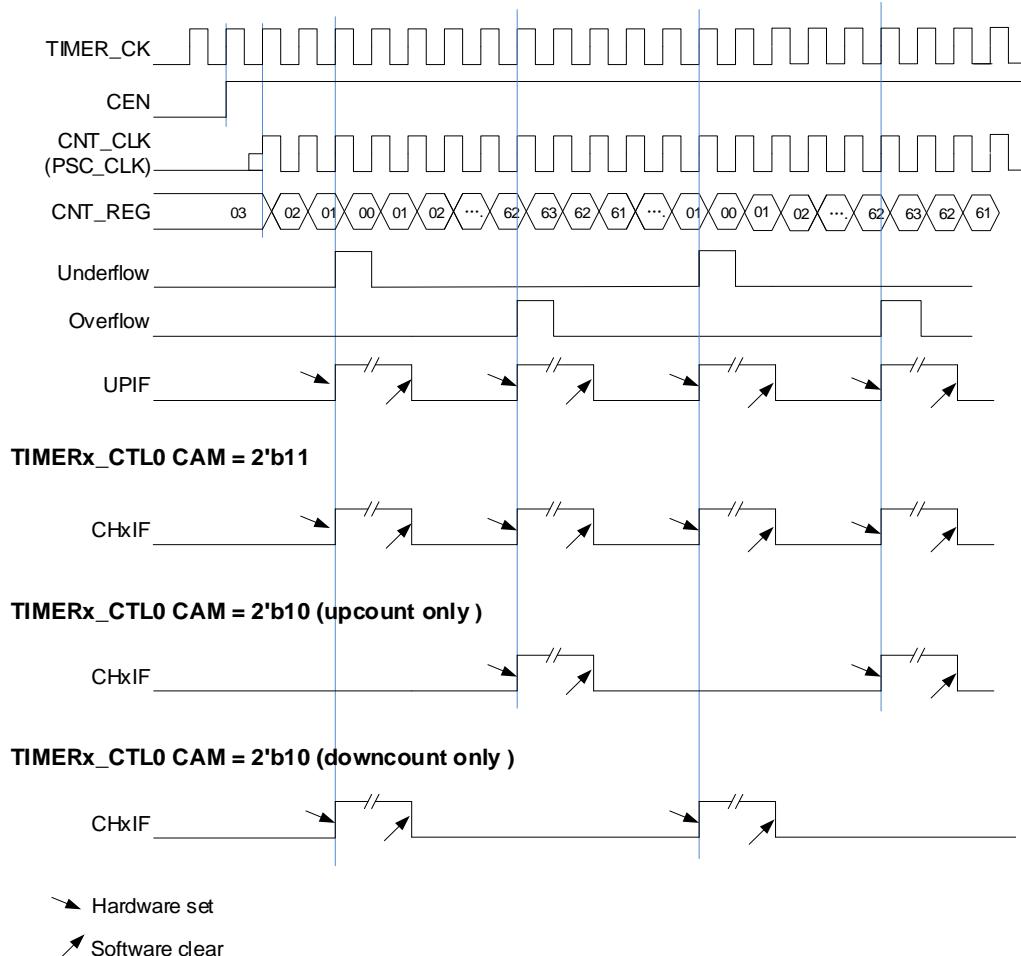
If set the UPDIS bit in the TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto-reload register,

prescaler register) are updated.

Figure 17-8. Timing chart of center-aligned counting mode show some examples of the counter behavior when TIMERx_CAR=0x63. TIMERx_PSC=0x0

Figure 17-8. Timing chart of center-aligned counting mode



Counter repetition

Counter Repetition is used to generate update event or updates the timer registers only after a given number ($N+1$) of cycles of the counter, where N is CREP 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 or at each counter overflow and at each counter underflow in center-aligned mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generate an update event.

For odd values of CREP in center-aligned mode, 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 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 17-9. Repetition counter timing chart of center-aligned counting mode

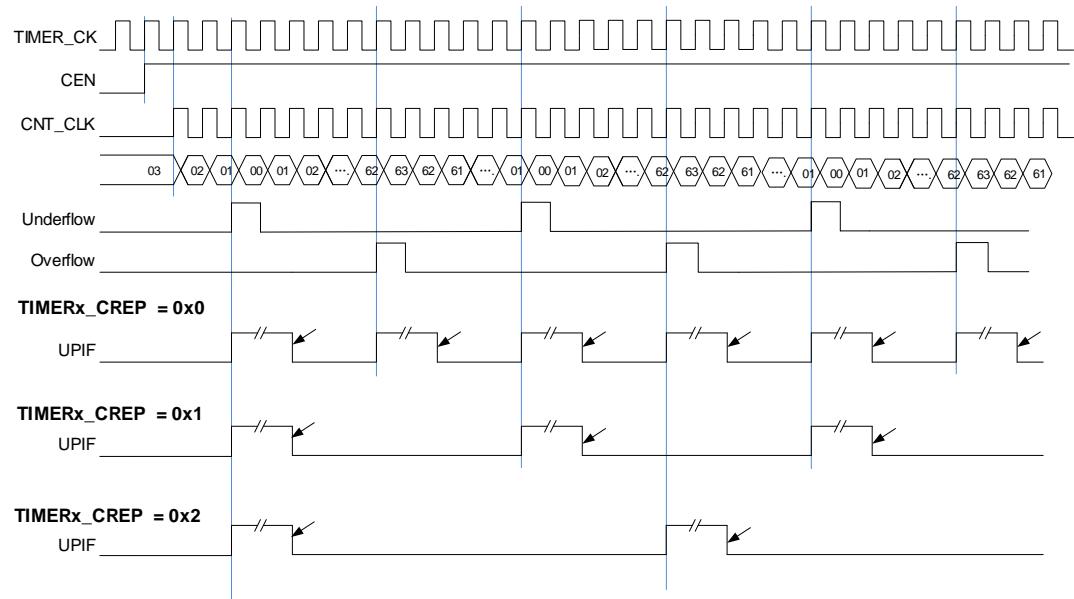


Figure 17-10. Repetition counter timing chart of up counting mode

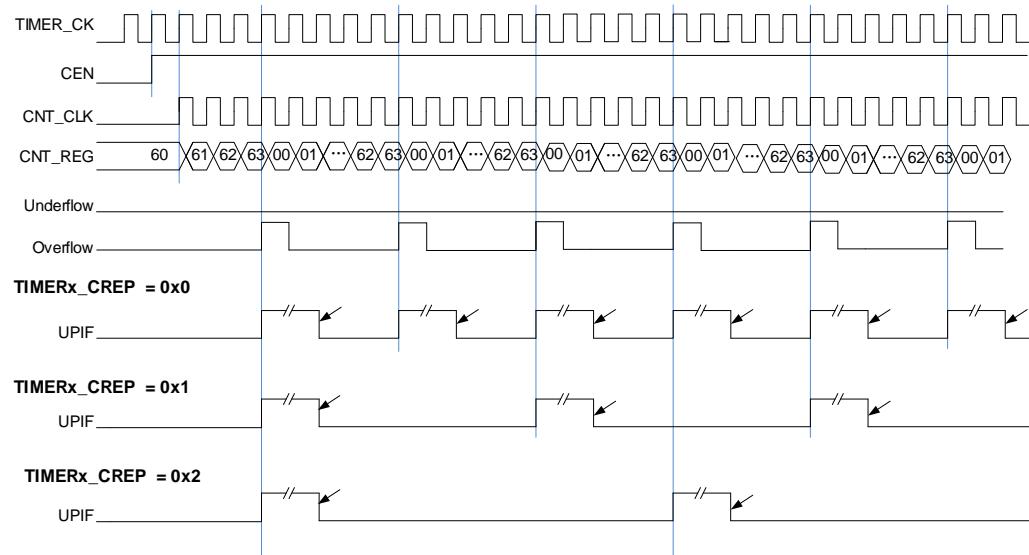
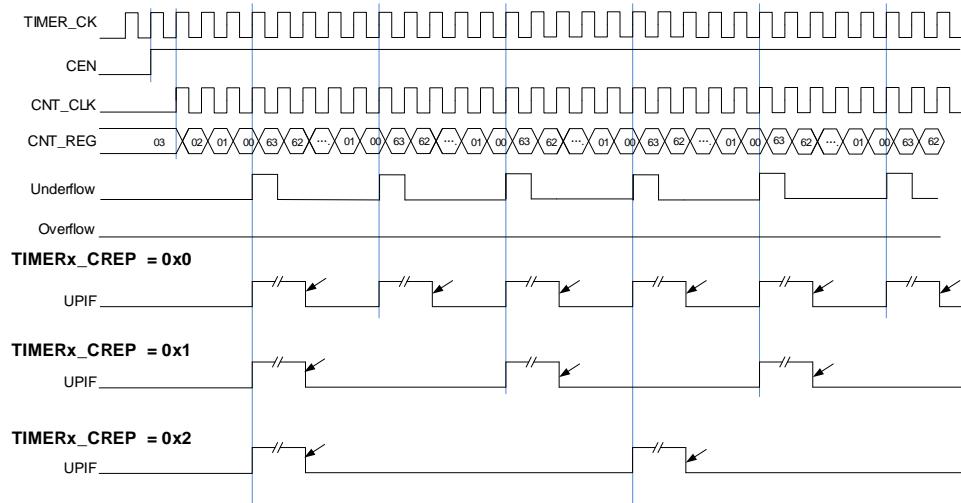


Figure 17-11. Repetition counter timing chart of down counting mode



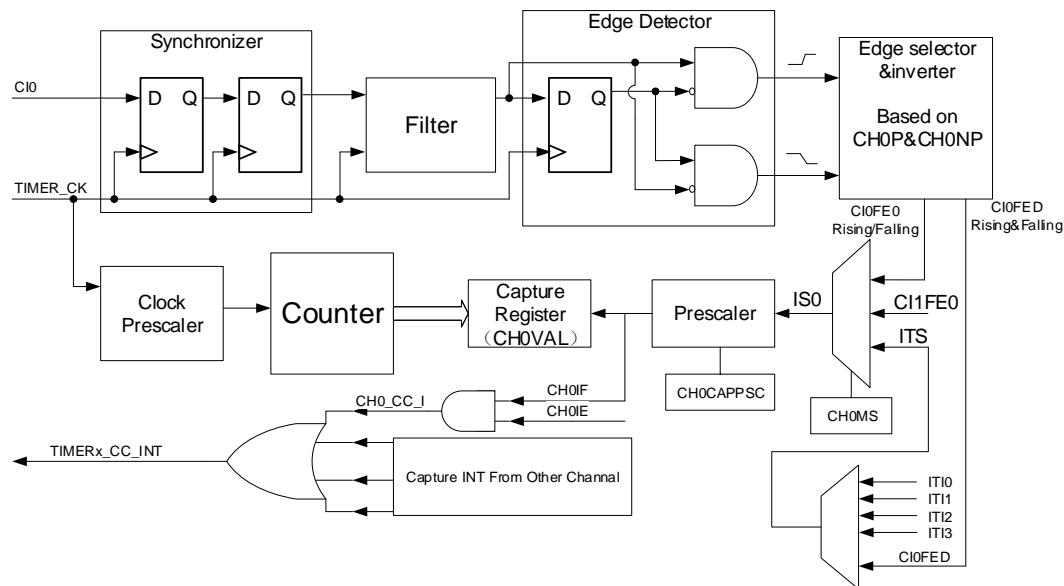
Capture/compare channels

The advanced 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.

■ Input capture mode

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 enabled by **CHxIE** = 1.

Figure 17-12. Input capture logic



One of channels' input signals (Clx) can be chosen from the TIMERx_CHx signal or the Exclusive-OR function of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the channel input signal (Clx) is synchronized to TIMER_CK 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 by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, TIMERx_CHxCV will restore the value of counter.

So, the process can be divided to several steps as below:

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel to input mode (CHxMS != 0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable. (CHxIE and CHxDEN in TIMERx_DMAINTEN)

Enable the related interrupt enable; you can get the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: when you wanted input signal is got, TIMERx_CHxCV will be set by counter's value.

And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt and DMA request will be asserted based on the configuration of CHxIE and CHxDEN in TIMERx_DMINTEN

Direct generation: if you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

■ Output compare mode

Figure 17-13. Output compare logic (with complementary output, x=0,1,2)

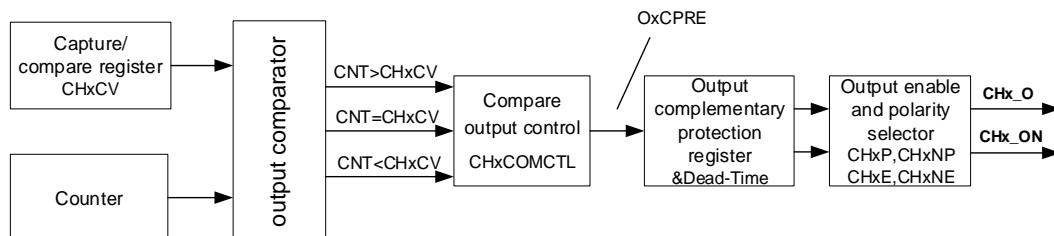
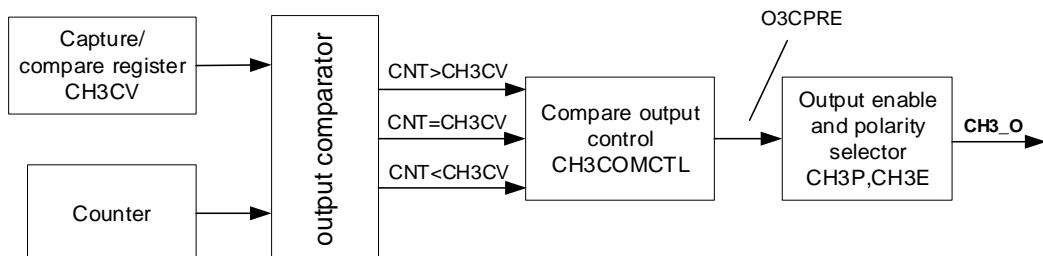


Figure 17-14. Output compare logic (CH3_O)



[Figure 17-13. Output compare logic \(with complementary output, x=0,1,2\)](#) and [Figure 17-14. Output compare logic \(CH3_O\)](#) show the logic circuit of output compare mode. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal (more details refer to [Channel output reference signal](#)) is described as below: The active level of O0CPRE is high, the output level of CH0_O/CH0_ON depends on OxCPRE signal, CHxP/CHxNP bit and CHxE/CHxNE bit (please refer to the TIMERx_CHCTL2 register for more details). For examples,

- 1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=1

(the output of CH_x_O is enabled),

If the output of OxCPRE is active(high) level, the output of CH_x_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CH_x_O is active(low) level.

- 2) Configure CH_xNP=0 (the active level of CH_x_ON is low, contrary to OxCPRE), CH_xNE=1 (the output of CH_x_ON is enabled),
 - If the output of OxCPRE is active(high) level, the output of CH_x_O is active(low) level;
 - If the output of OxCPRE is inactive(low) level, the output of CH_x_O is active(high) level.

When CH₀_O and CH₀_ON are output at the same time, the specific outputs of CH₀_O and CH₀_ON are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMER_x_CCHP register. Please refer to [Outputs complementary](#) for more details.

In output compare mode, the TIMER_x can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMER_x_CH_xCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CH_xCOMCTL. When the counter reaches the value in the TIMER_x_CH_xCV register, the CH_xIF bit is set and the channel (n) interrupt is generated if CH_xIE = 1. And the DMA request will be asserted, if CxCDE=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 shadow enable mode by CH_xCOMSEN
- Set the output mode (Set/Clear/Toggle) by CH_xCOMCTL.
- Select the active high polarity by CH_xP/CH_xNP
- Enable the output by CH_xEN

Step3: Interrupt/DMA-request enables configuration by CH_xIE/CxCDE

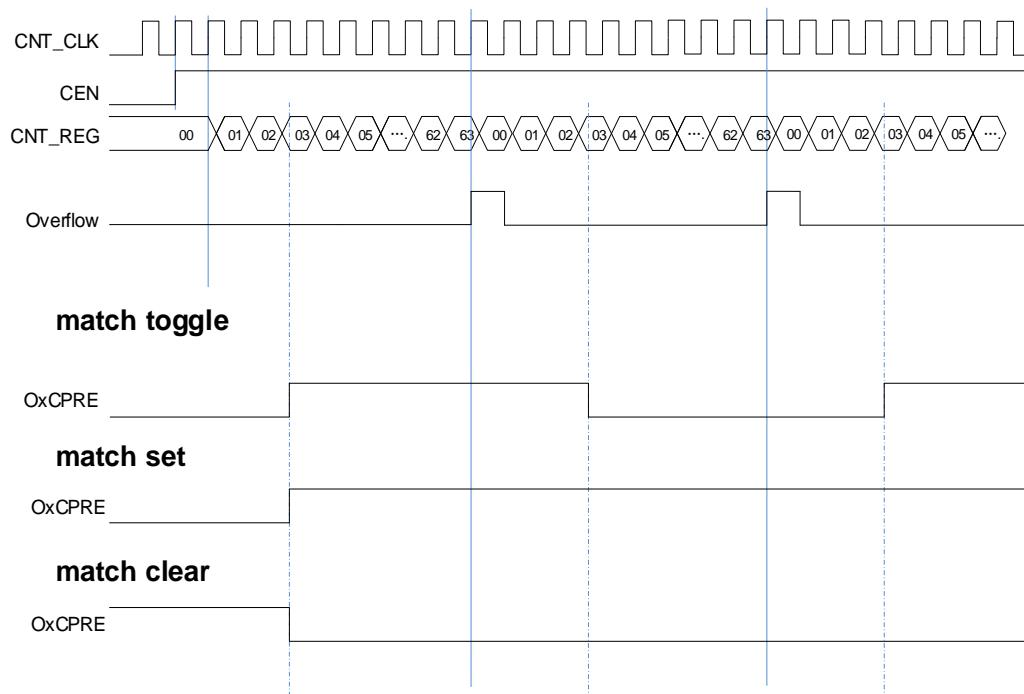
Step4: Compare output timing configuration by TIMER_x_CAR and TIMER_x_CH_xCV

About the TIMER_x_CH_xCV; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

[Figure 17-15. Output-compare in three modes](#) show the three compare modes toggle/set/clear. CAR=0x63, CH_xVAL=0x3

Figure 17-15. Output-compare in three modes



PWM mode

In the output PWM mode (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, we can also divide PWM into EAPWM (Edge aligned PWM) and CAPWM (Centre aligned PWM).

The EAPWM period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. [Figure 17-16. Timing chart of EAPWM](#) shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by 2*TIMERx_CAR, and duty cycle is by 2*TIMERx_CHxCV. [Figure 17-17. Timing chart of CAPWM](#) shows the CAPWM output and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).

And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 17-16. Timing chart of EAPWM

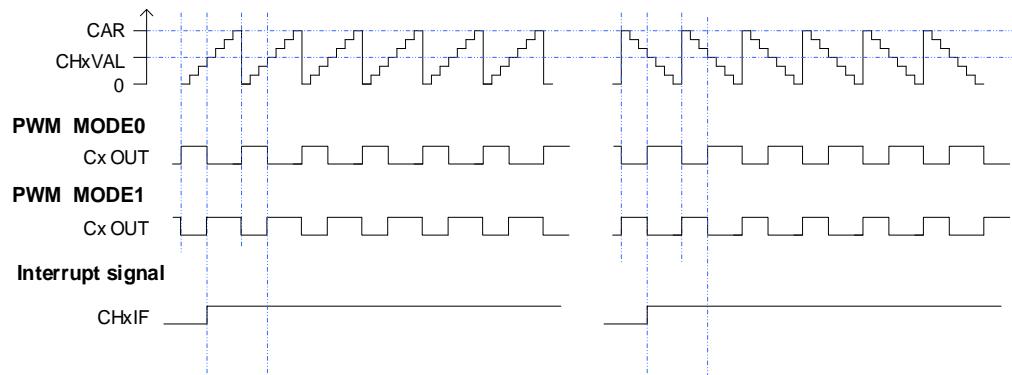
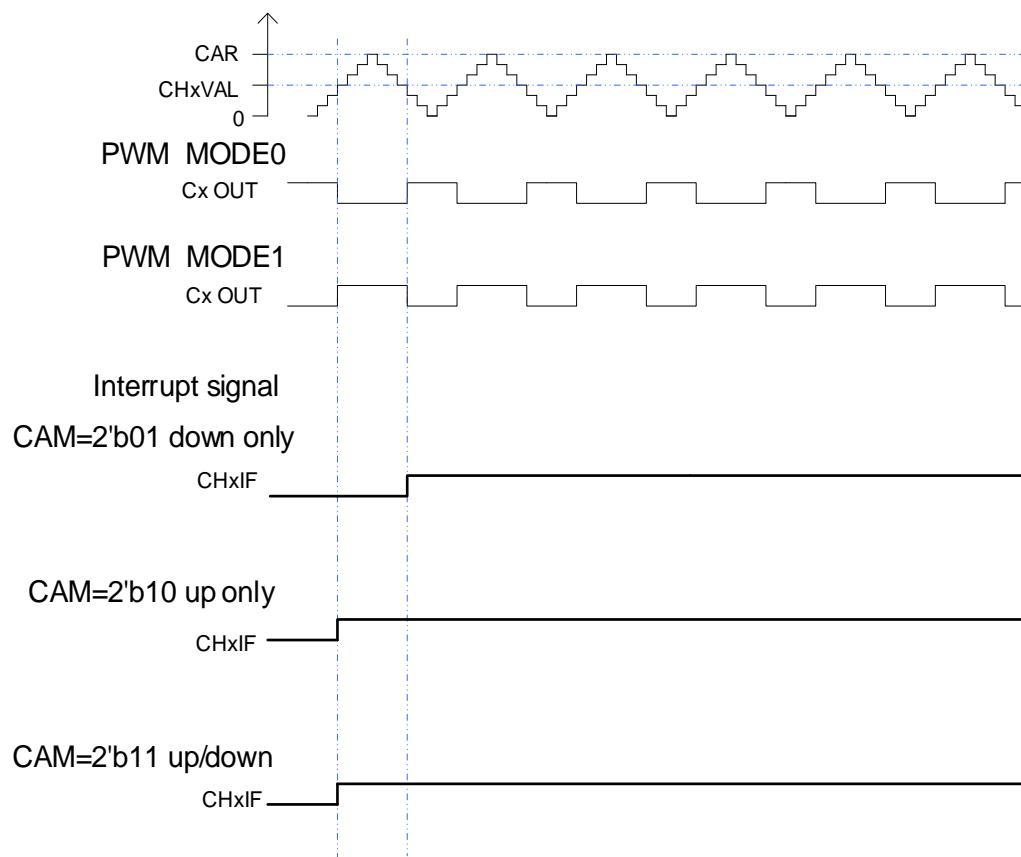


Figure 17-17. Timing chart of CAPWM



Channel output reference signal

As is shown in [Figure 17-13. Output compare logic \(with complementary output, \$x=0, 1, 2\$ \)](#), when the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL field. The OxCPRE signal has several types of output function. These include, keeping the original level by setting

the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

The OxCPRE signal can be forced to 0 when the ETIFE signal is derived from the external ETI pin and when it is set to a high level by setting the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register. The OxCPRE signal will not return to its active level until the next update event occurs.

Outputs complementary

Function of complementary is for a pair of CHx_O and CHx_ON. Those two output signals cannot be active at the same time. The TIMERx has 4 channels, but only the first three channels have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 17-2. Complementary outputs controlled by parameters

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable.	
				1		
			1	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output disable. If clock is enable: CHx_O = ISOx CHx_ON = ISOxN	
				1		
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output disable.	
				1		
			1	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output enable. If clock is enable: CHx_O = ISOx CHx_ON = ISOxN	
				1		
1	0/1	0	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON output disable.	
				1	CHx_O = LOW CHx_O output disable.	CHx_ON=OxCPR E⊕CHx NP CHx_ON output enable
			1	0	CHx_O=OxCPR E⊕CHx P CHx_O output enable	CHx_ON = LOW CHx_ON output disable.
				1	CHx_O=OxCPR E⊕CHx P CHx_O output enable	CHx_ON=(!OxCPR E)⊕ CHx NP CHx_ON output enable
			0	0	CHx_O = CHxP CHx_O output disable.	CHx_ON = CHxNP CHx_ON output disable.
				1	CHx_O = CHxP CHx_O output enable	CHx_ON=OxCPR E⊕CHx NP CHx_ON output enable
		1	0	0	CHx_O=OxCPR E⊕CHx P CHx_O output enable	CHx_ON = CHxNP CHx_ON output enable.
				1	CHx_O=OxCPR E⊕CHx P CHx_O output enable	CHx_ON=(!OxCPR E)⊕ CHx NP CHx_ON output enable.

Dead time insertion

The dead time insertion is enabled when both CHxEN and CHxNEN are 1'b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for all channels expect for channel 3. The detail about the delay time, refer to the register TIMERx_CCHP.

The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

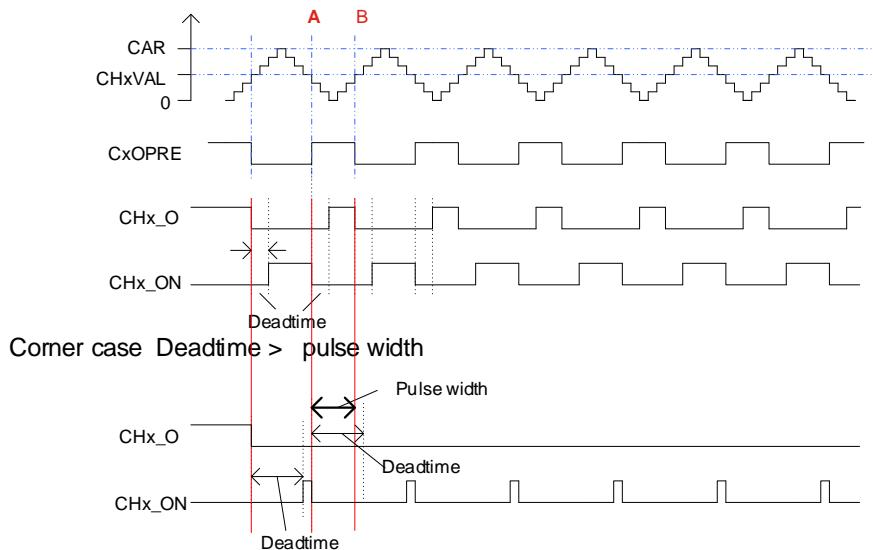
When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the [Figure 17-18. Complementary output with dead-time insertion](#), CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, at point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value 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 CHx_O duty cycle, then the CHx_O signal is always the inactive value. (As show in the [Figure 17-18. Complementary output with dead-time insertion](#).)

- The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 17-18. Complementary output with dead-time insertion.



Break function

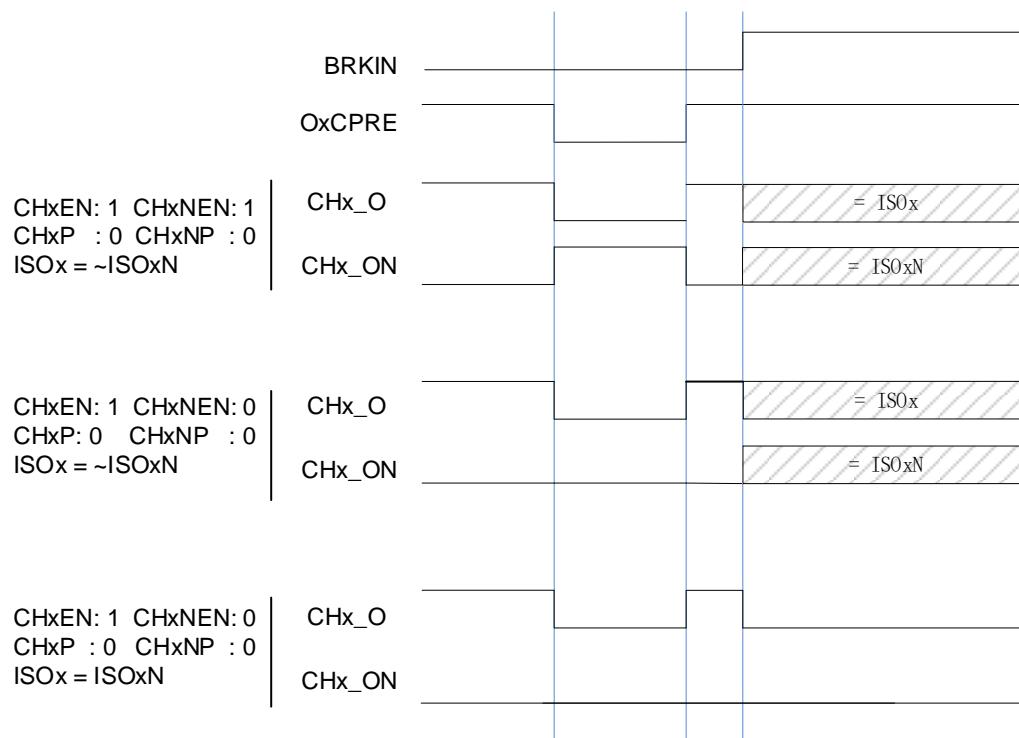
In this function, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs. The break sources are input break pin and HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the

TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead-time.

When a break occurs, the BRKIF bit in the TIMERx_INTF register is set. If BRKIE is 1, an interrupt generated.

Figure 17-19. Output behavior of the channel in response to a break (the break high active)



Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting SMC=0x01, 0x02, or 0x03 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 17-3. Counting direction versus encoder signals](#). 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 TIMERx_CAR register before the counter starts to count.

Table 17-3. Counting direction versus encoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Cl0 only counting	Cl1FE1=High	Dow n	Up	-	-
	Cl1FE1=Low	Up	Dow n	-	-
Cl1 only counting	Cl0FE0=High	-	-	Up	Dow n
	Cl0FE0=Low	-	-	Dow n	Up
Cl0 and Cl1 counting	Cl1FE1=High	Dow n	Up	X	X
	Cl1FE1=Low	Up	Dow n	X	X
	Cl0FE0=High	X	X	Up	Dow n
	Cl0FE0=Low	X	X	Dow n	Up

Note: "-" means "no counting"; "X" means impossible.

Figure 17-20. Example of counter operation in encoder interface mode

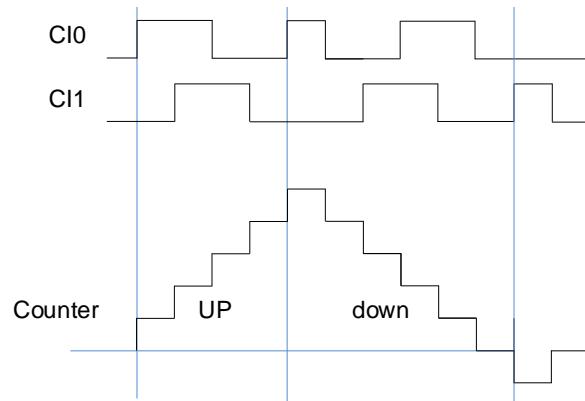
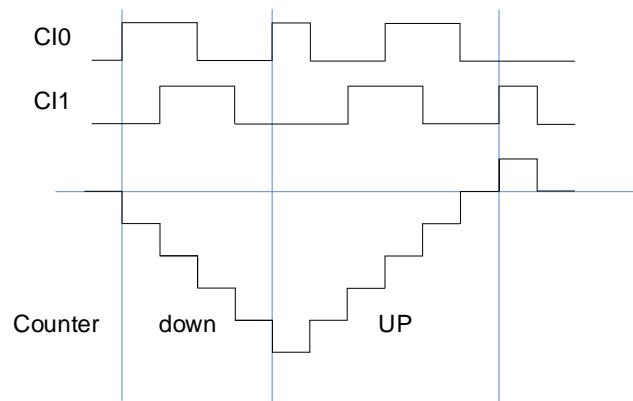


Figure 17-21. Example of encoder interface mode with Cl0FE0 polarity inverted



Hall sensor function

Hall sensor is generally used to control BLDC Motor; advanced timer can support this function.

[Figure 17-22. Hall sensor is used to BLDC motor](#) show how to connect. And we can see we need two timers. First TIMER_in (Advanced/General L0 TIMER) should accept three Rotor Position signals from Motor.

Each of the 3 sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced.

By the internal connection such as TRGO-ITIx, TIMER_in and TIMER_out can be connected. TIMER_out will generate PWM signal to control BLDC motor's speed based on the ITRx. Then, the feedback circuit is finished, also you change configuration to fit your request.

About the TIMER_in, it need have input XOR function, so you can choose from Advanced/General L0 TIMER.

And TIMER_out need have functions of complementary and Dead-time, so only advanced timer can be chosen. Else, based on the timers' internal connection relationship, pair's timers can be selected. For example:

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

After getting appropriate timers combination, and wire connection, we need to configure timers. Some key settings include:

- Enable XOR by setting TI0S, then, each of input signal change will make the CI0 toggle. CH0VAL will record the value of counter at that moment.
- Enable ITIx connected to commutation function directly by setting CCUC and CCSE.
- Configuration PWM parameter based on your request.

Figure 17-22. Hall sensor is used to BLDC motor

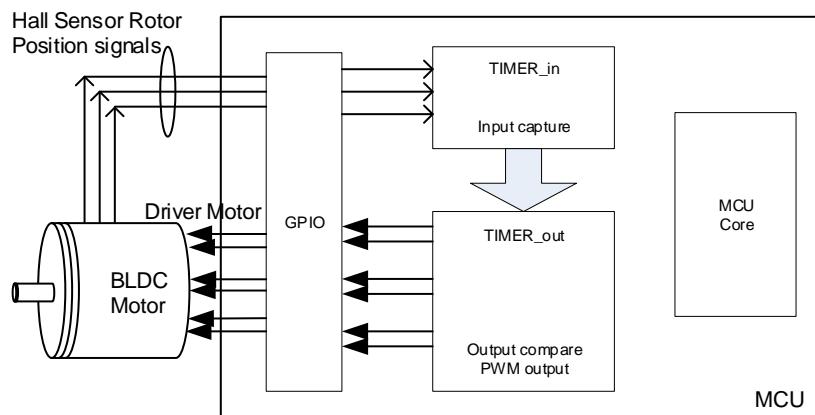
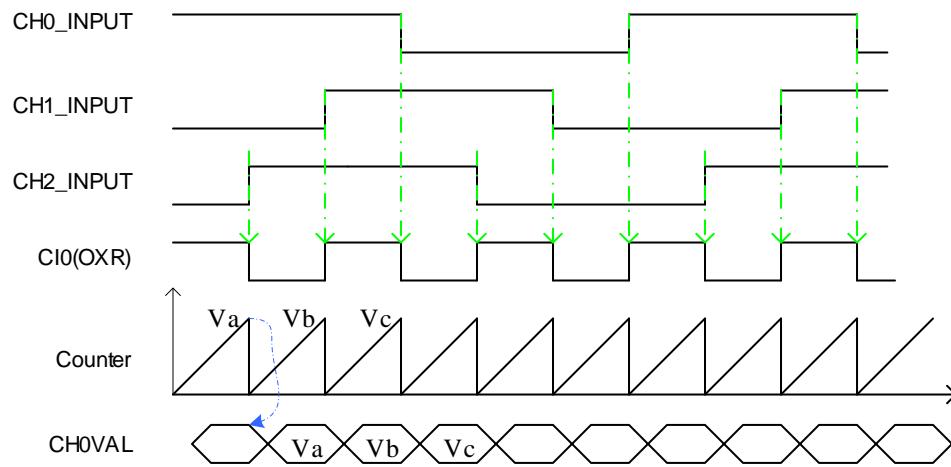
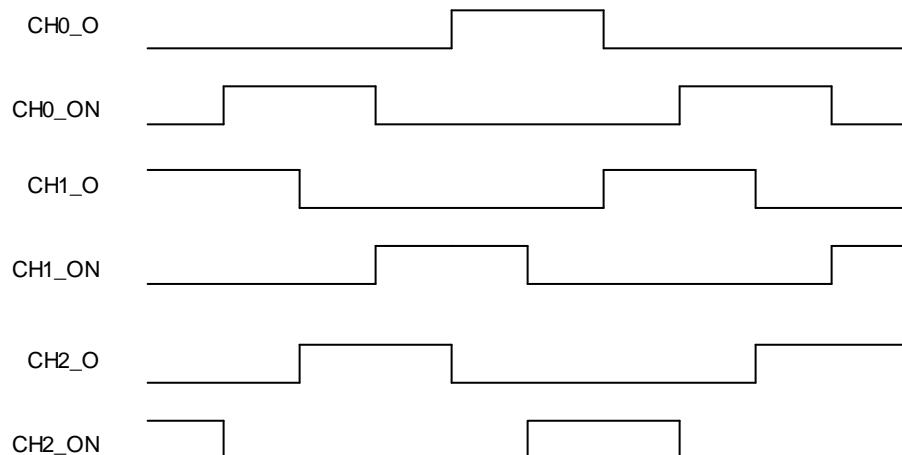


Figure 17-23. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



Advanced TIMER_out under output compare mode(PWM with Dead-time)

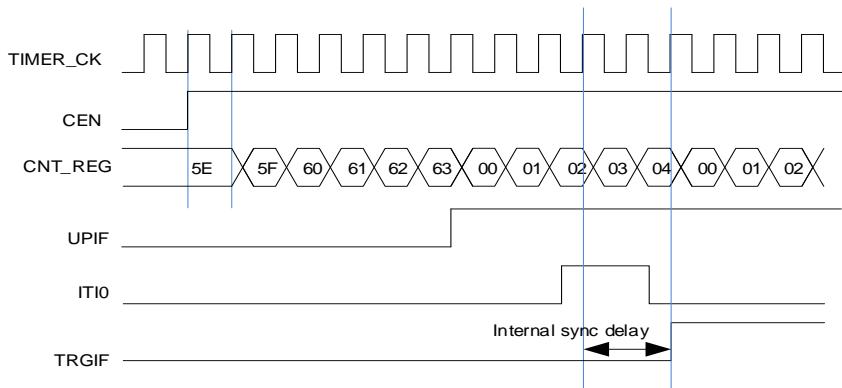
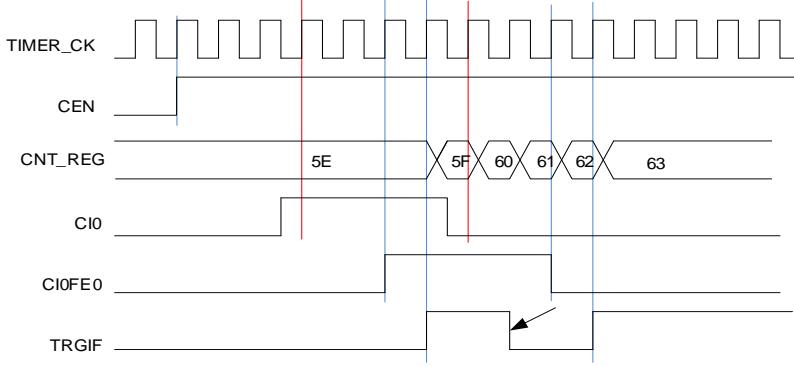


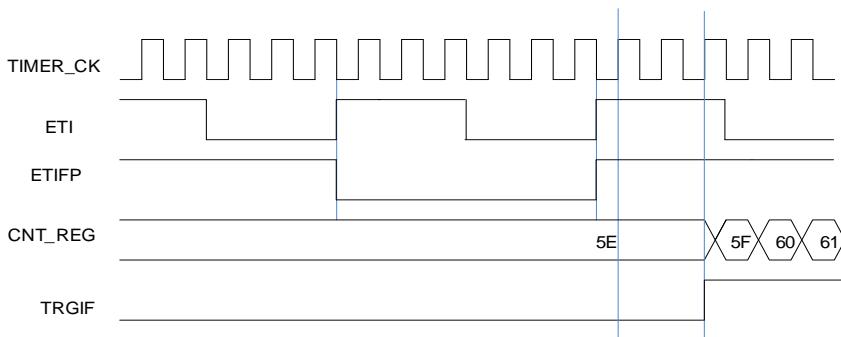
Slave controller

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode which is selected by the SMC [2:0] in the TIMERx_SMCFG register. The trigger input of these modes can be selected by the TRGS [2:0] in the TIMERx_SMCFG register.

Table 17-4. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[2:0]	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion.	For the ITIx, no filter and prescaler can be used.
		000: ITI0		
		001: ITI1		
		010: ITI2		
		011: ITI3		For the Clx, filter can be used by configuring CHxCAPFLT, no
		100: CI0F_ED		

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
		101: CI0FE0 110: CI1FE1 111: ETIFP	If ETIFP is selected as the trigger source, configure the ETP for polarity selection and inversion.	prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
	Restart mode The counter will be cleared and restart when a rising edge of trigger input comes.	TRGS[2:0] = 3'b000 ITIO is selected.	For ITIO, no polarity selector can be used.	For the ITIO, no filter and prescaler can be used.
Figure 17-24. Restart mode				
Exam 1				 <p>The diagram shows the following waveforms:</p> <ul style="list-style-type: none"> TIMER_CK: A square wave clock signal. CEN: A pulse signal that enables the timer. CNT_REG: A counter register showing values: 5E, 5F, 60, 61, 62, 63, 00, 01, 02, 03, 04, 00, 01, 02. UPIF: An update flag signal. ITIO: A trigger input signal that triggers a counter reset and start. TRGIF: A trigger interrupt flag signal. <p>A note indicates an "Internal sync delay" between the ITIO rising edge and the TRGIF signal.</p>
	Pause mode The counter will be paused when the trigger input is low, and it will start when the trigger input is high.	TRGS[2:0]=3'b101 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.
Figure 17-25. Pause mode				
Exam 2				 <p>The diagram shows the following waveforms:</p> <ul style="list-style-type: none"> TIMER_CK: A square wave clock signal. CEN: A pulse signal that enables the timer. CNT_REG: A counter register showing values: 5E, 5F, 60, 61, 62, 63. CI0: A capture input signal. CI0FE0: A trigger input signal that controls the pause mode. TRGIF: A trigger interrupt flag signal. <p>Red vertical lines mark the capture event at the rising edge of CI0 and the pause/start transition at the rising edge of CI0FE0.</p>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Event mode The counter will start to count when a rising edge of trigger input comes.	TRGS[2:0] =3'b111 ETIFFP 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 17-26. Event mode				
Exam3		 <p>The diagram shows the following signals over time:</p> <ul style="list-style-type: none"> TIMER_CK: A square wave clock signal. ETI: An external trigger input signal that generates a pulse whenever it transitions from low to high. ETIFFP: The filtered trigger signal, which is high during the pulse from ETI and remains high until the next update event. CNT_REG: The counter value, which increments every clock cycle of TIMER_CK while ETIFFP is high. It is shown with values 5E, 5F, 60, and 61. TRGIF: The trigger interrupt flag, which is set high when ETIFFP goes high and remains high until the next update event. 		

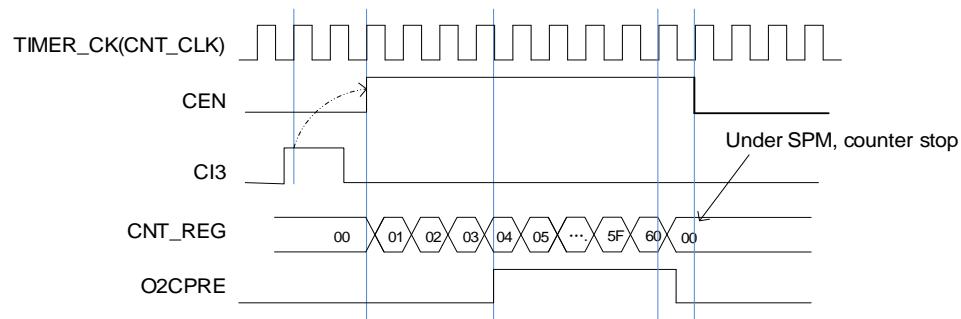
Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event automatically. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the CEN bit to 1 using software. Setting the CEN bit to 1 or a trigger from the trigger signals 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 using software, the counter will be stopped and its value 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 trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0/1 register. 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. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 17-27. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60

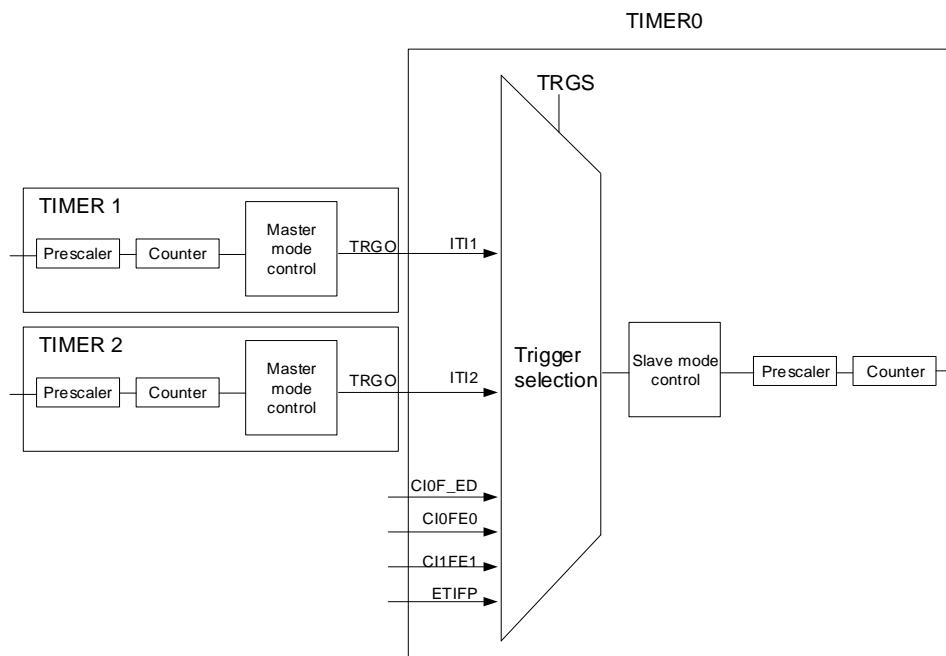


Timers interconnection

The timers can be internally connected together 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 present several examples of trigger selection for the master and slave modes.

Figure 17-28. Timer0 master/slave mode example shows the timer0 trigger selection when it is configured in slave mode.

Figure 17-28. Timer0 master/slave mode example



Other interconnection examples:

- **TIMER2** as the prescaler for **TIMER0**

TIMER2 is configured as a prescaler for TIMER0. Refer to [Figure 17-28. Timer0 master/slave mode example for connections](#). Steps are shown as follows:

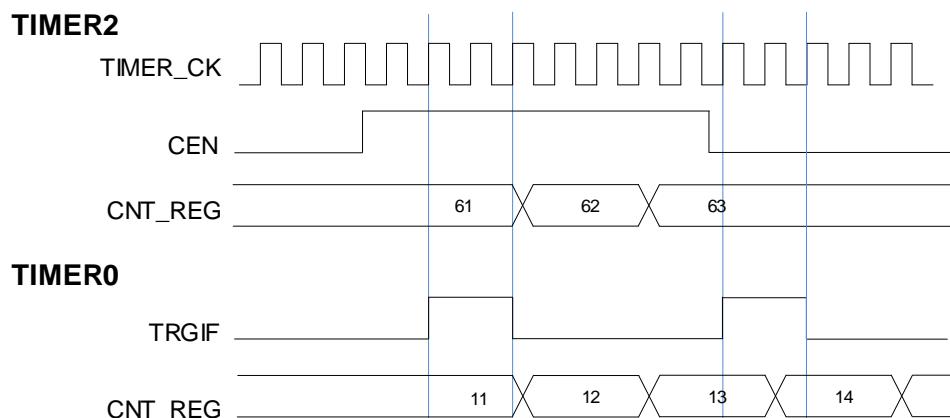
1. Configure TIMER2 in master mode and select its update event (UPE) as trigger output (MMC=3'b010 in the TIMER2_CTL1 register). Then TIMER2 drives a periodic signal on each counter overflow.
 2. Configure TIMER2 period (TIMER2_CAR register).
 3. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
 4. Configure TIMER0 in external clock mode 0 (SMC=3'b111 in TIMERx_SMCFG register).
 5. Start TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
 6. 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 17-29. 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. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
3. Configure TIMER0 in event mode (SMC=3'b 110 in TIMERx_SMCFG register).
4. Start TIMER2 by writing 1 to the CEN bit (TIMER2_CTL0 register).

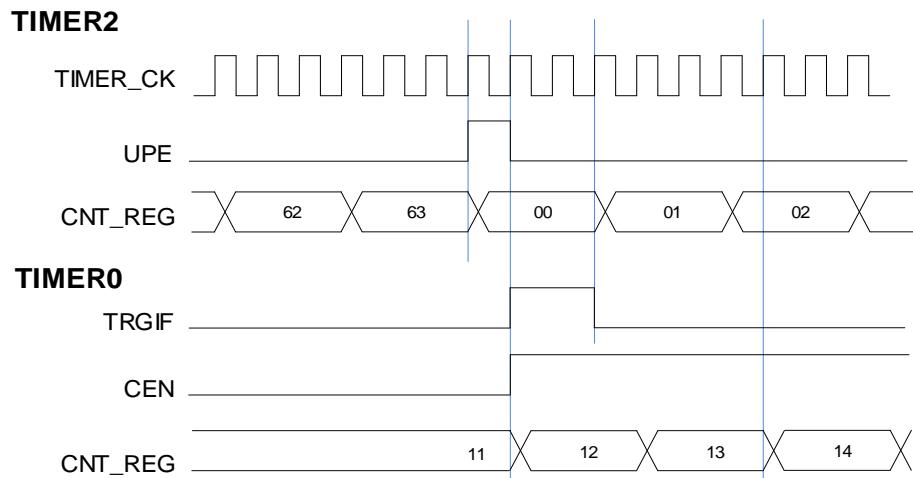
Figure 17-29. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to [Figure 17-30. 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. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
4. Configure TIMER0 in event mode (SMC=3'b110 in TIMERx_SMCFG register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 17-30. Trigger mode of TIMER0 controlled by update signal of TIMER2

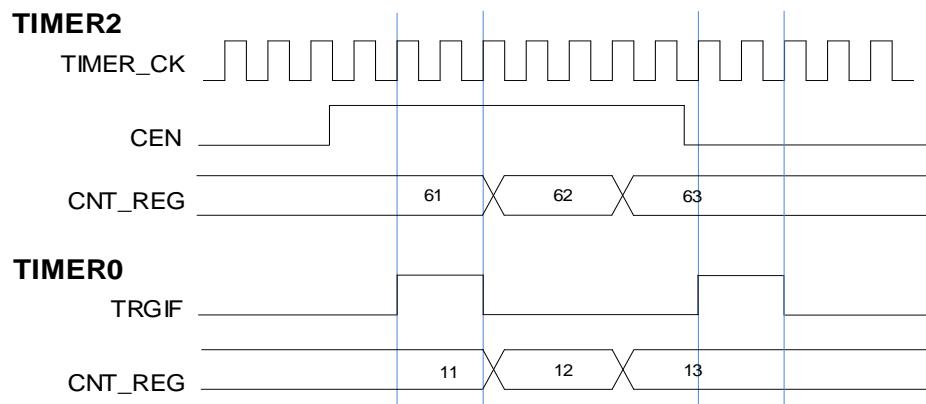


- Enable TIMER0 to count with the enable/O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to [Figure 17-31. 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. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
3. Configure TIMER0 in pause mode (SMC=3'b101 in TIMERx_SMCFG 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).
6. Stop TIMER2 by writing '0' to the CEN bit (TIMER2_CTL0 register).

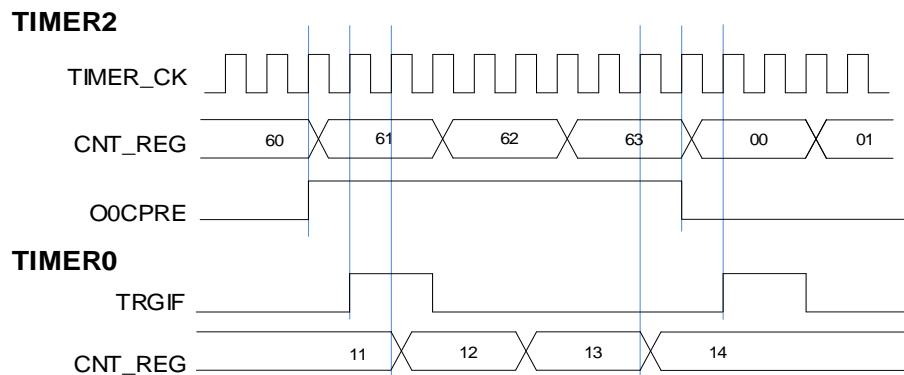
Figure 17-31. Pause mode of TIMER0 controlled by enable signal of TIMER2



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 (MMS=3'b100 in the TIMER2_CTL1 register).
2. Configure the TIMER2 O0CPRE waveform (TIMER2_CHCTL0 register).
3. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
4. Configure TIMER0 in pause mode (SMC=3'b101 in TIMERx_SMCFG register).
5. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
6. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 17-32. Pause mode of TIMER0 controlled by O0CPREF signal of TIMER2



- Using an external trigger to start two timers synchronously.

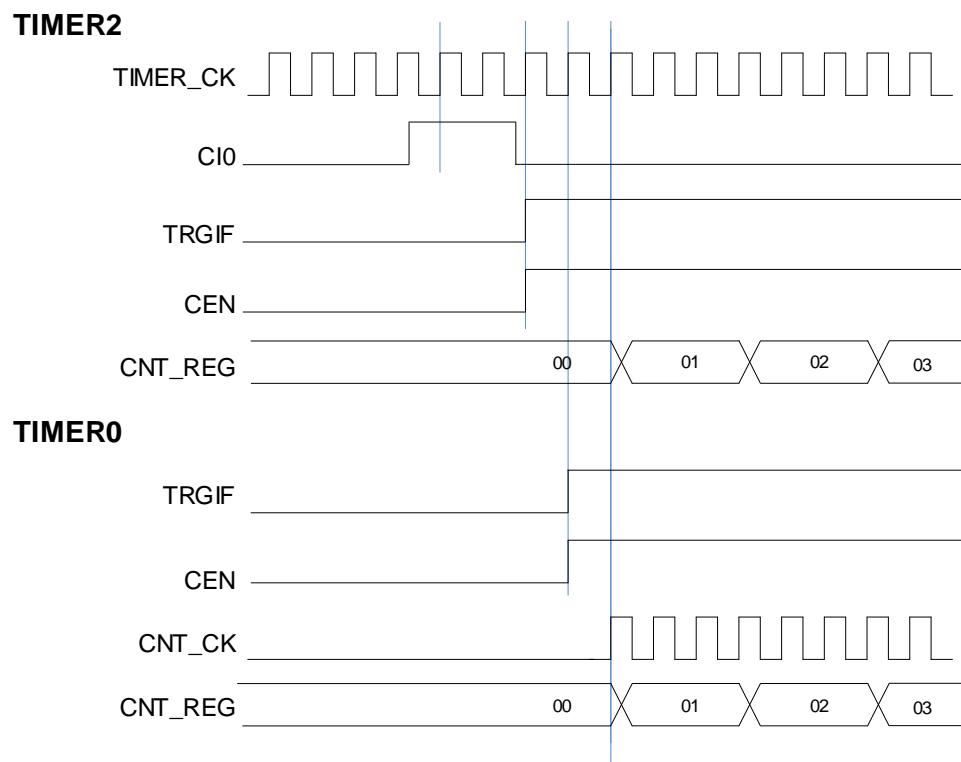
The start of TIMER0 is triggered by the enable signal of TIMER2, and TIMER2 is triggered by its Cl0 input rising edge. To ensure that two timers start synchronously, TIMER2 must be configured in master/slave mode. Steps are shown as follows:

1. Configure TIMER2 in slave mode, and select Cl0_ED as the input trigger (TRGS=3'b100 in the TIMER2_SMCFG register).
2. Configure TIMER2 in event mode (SMC=3'b110 in the TIMER2_SMCFG register).

3. Configure TIMER2 in master/slave mode by writing MSM=1 (TIMER2_SMCFG register).
4. Select TIMER2 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
5. Configure TIMER0 in event mode (SMC=3'b110 in the TIMER0_SMCFG 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 17-33. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



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.

Timer debug mode

When the Cortex™-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register is set to 1, the TIMERx counter stops.

17.1.5. **TIMERx registers(x=0)**

TIMER0 secure access base address: 0x5001 0000

TIMER0 non-secure access base address: 0x4001 0000

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																				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved					CKDIV[1:0]		ARSE		CAM[1:0]		DIR		SPM		UPS		UPDIS		CEN	
					rw		rw		rw		rw		rw		rw		rw			

Bits	Fields	Descriptions
31:10	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 the timer clock (TIMER_CK) and the dead-time and sampling clock (DTS), which is used by the dead-time generators and the digital filters. 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 TIMERx_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 under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_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 under center-aligned and channel is configured in output mode (CHxMS=00 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 under center-aligned and channel is configured in output mode (CHxMS=00 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, cannot be switched from 0x00 to non 0x00.
4	DIR	<p>Direction</p> <p>0: Count up</p> <p>1: Count down</p> <p>This bit is read only when the timer is configured in center-aligned mode or encoder mode.</p>
3	SPM	<p>Single pulse mode.</p> <p>0: Counter continues after update event.</p> <p>1: The CEN is cleared by hardware and the counter stops at next update event.</p>
2	UPS	<p>Update source</p> <p>This bit is used to select the update event sources by software.</p> <p>0: Any of the following events generate an update interrupt or DMA request:</p> <ul style="list-style-type: none"> The UPG bit is set The counter generates an overflow or underflow event The slave mode controller generates an update event. <p>1: Only counter overflow/underflow generates an update interrupt or DMA request.</p>
1	UPDIS	<p>Update disable.</p> <p>This bit is used to enable or disable the update event generation.</p> <p>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:</p> <ul style="list-style-type: none"> The UPG bit is set The counter generates an overflow or underflow event The slave mode controller generates an update event. <p>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 if the slave mode controller generates a hardware reset event.</p>
0	CEN	<p>Counter enable</p> <p>0: Counter disable</p> <p>1: Counter enable</p> <p>The CEN bit must be set by software when timer works in external clock, pause mode and encoder mode. While in event mode, the hardware can set the CEN bit automatically.</p>

Control register 1 (TIMERx_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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISOON	ISO0	TIOS	MMC[2:0]	DMAS	CCUC	Reserved	CCSE		

rw rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	ISO3	Idle state of channel 3 output Refer to ISO0 bit
13	ISO2N	Idle state of channel 2 complementary output Refer to ISOON bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of channel 1 complementary output Refer to ISOON bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISOON	Idle state of channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON 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 CH0_ON is implemented. This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
7	TIOS	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, CH1 and 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 in master mode to slave timers 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 useful to start several timers at the same time or to control a window in which a slave timer is enabled. 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 occurred in channel0.

100: Compare. In this mode the master mode controller selects the O0CPRE signal is used as TRGO

101: Compare. In this mode the master mode controller selects the O1CPRE signal is used as TRGO

110: Compare. In this mode the master mode controller selects the O2CPRE signal is used as TRGO

111: Compare. In this mode the master mode controller selects the O3CPRE signal is used as TRGO

3	DMAS	DMA request source selection 0: DMA request of channel x is sent when capture/compare event occurs. 1: DMA request of channel x is sent when update event occurs.
2	CCUC	Commutation control shadow register update control When the commutation control shadow enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below : 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by when CMTG bit is set or a rising edge of TRGI occurs. 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, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. 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).

31	30	29	28	27	26	25	24	23	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		TRGS[2:0]	Reserved		SMC[2:0]					
rw	rw	rw		rw		rw		rw		rw					rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	<p>External trigger polarity</p> <p>This bit specifies the polarity of ETI signal</p> <p>0: ETI is active at high level or rising edge.</p> <p>1: ETI is active at low level or falling edge.</p>
14	SMC1	<p>Part of SMC for enable External clock mode1.</p> <p>In external clock mode 1, the counter is clocked by any active edge on the ETIF signal.</p> <p>0: External clock mode 1 disabled</p> <p>1: External clock mode 1 enabled.</p> <p>It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TRGS bits must not be 3'b111 in this case.</p> <p>The external clock input will be ETIF if external clock mode 0 and external clock mode 1 are enabled at the same time.</p> <p>Note: External clock mode 0 enable is in this register's SMC bit-field.</p>
13:12	ETPSC[1:0]	<p>External trigger prescaler</p> <p>The frequency of external trigger signal ETI must not be at higher than 1/4 of TIMER_CK frequency. When the external trigger signal is a fast clock, the prescaler can be enabled to reduce ETI frequency.</p> <p>00: Prescaler disable</p> <p>01: ETI frequency will be divided by 2</p> <p>10: ETI frequency will be divided by 4</p> <p>11: ETI frequency will be divided by 8</p>
11:8	ETFC[3:0]	<p>External trigger filter control</p> <p>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 ETI signal and the length of the digital filter applied to ETI.</p> <p>0000: Filter disabled. fSAMP=fDTS, N=1.</p> <p>0001: fSAMP=fCK_TIMER, N=2.</p> <p>0010: fSAMP=fCK_TIMER, N=4.</p>

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	<p>Master-slave mode</p> <p>This bit can be used to synchronize selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected together.</p> <p>0: Master-slave mode disable 1: Master-slave mode enable</p>
6:4	TRGS[2:0]	<p>Trigger selection</p> <p>This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter.</p> <p>000: Internal trigger input 0 (ITI0) 001: Internal trigger input 1 (ITI1) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: Cl0 edge flag (Cl0F_ED) 101: channel 0 input Filtered output (Cl0FE0) 110: channel 1 input Filtered output (Cl1FE1) 111: External trigger input filter output(ETIFFP)</p> <p>These bits must not be changed when slave mode is enabled.</p>
3	Reserved	Must be kept at reset value.
2:0	SMC[2:0]	<p>Slave mode control</p> <p>000: Disable mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high.</p> <p>001: Quadrature decoder mode 0.The counter counts on Cl1FE1 edge, while the direction depends on Cl0FE0 level.</p> <p>010: Quadrature decoder mode 1.The counter counts on Cl0FE0 edge, while the direction depends on Cl1FE1 level.</p> <p>011: Quadrature decoder mode 2.The counter counts on both Cl0FE0 and Cl1FE1 edge, while the direction depends on each other.</p> <p>100: Restart mode. The counter is reinitialized and the shadow registers are</p>

updated on the rising edge of the selected trigger input.

101: Pause mode. The trigger input enables the counter clock when it is high and disables the counter when it is low.

110: Event mode. A rising edge of the trigger input enables the counter. The counter cannot be disabled by the slave mode controller.

111: External clock mode 0. The counter counts on the rising edges of the selected trigger.

DMA and interrupt enable register (TIMERx_DMINTEN)

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
Reserved	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CHOIE	UPIE
rw	rw	rw	rw	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	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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	CH3OF	CH2OF	CH1OF	CH0OF	Reserved	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CHOIF	UPIF
	rc_w0	rc_w0	rc_w0	rc_w0	.	rc_w0							

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	Reserved	Must be kept at reset value.
7	BRKIF	Break interrupt flag This flag is set by hardware when the break input goes active, and cleared by software if the break input is not active. 0: No active level break has been detected. 1: An active level 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 on trigger input generates a trigger event. When the slave mode controller is enabled in pause mode both edges on trigger input generates 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 channel's commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3's capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2's capture/compare interrupt flag

		Refer to CH0IF description
2	CH1IF	Channel 1's capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0's 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. If Channel0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No Channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_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
Reserved															
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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	BRKG	Break event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is 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 TIMERx_INTF register is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event

		1: Generate a trigger event
5	CMTG	<p>Channel commutation event generation</p> <p>This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1).</p>
		0: No affect
		1: Generate channel's c/c control update event
4	CH3G	<p>Channel 3's capture or compare event generation</p> <p>Refer to CH0G description</p>
3	CH2G	<p>Channel 2's capture or compare event generation</p> <p>Refer to CH0G description</p>
2	CH1G	<p>Channel 1's capture or compare event generation</p> <p>Refer to CH0G description</p>
1	CH0G	<p>Channel 0's capture or compare event generation</p> <p>This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high.</p> <p>0: No generate a channel 1 capture or compare event</p> <p>1: Generate a channel 1 capture or compare event</p>
0	UPG	<p>Update event generation</p> <p>This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time.</p> <p>0: No generate an update event</p> <p>1: Generate an update event</p>

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															
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]		

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 description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMSEN description
9:8	CH1MS[1:0]	Channel 1 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. (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 TRGS bits in TIMERx_SMCFG 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 ETIF input. 0: Channel 0 output compare clear disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of the output reference signal O0CPRE which drives CH0_O and CH0_ON. O0CPRE is active high, while CH0_O and CH0_ON active level depends on CH0P and CH0NP bits. 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. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV.

		<p>011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV.</p> <p>100: Force low. O0CPRE is forced low level.</p> <p>101: Force high. O0CPRE is forced high level.</p> <p>110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active.</p> <p>111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive.</p> <p>When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from “Timing mode” mode to “PWM” mode or when the result of the comparison changes.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 00(COMPARE MODE).</p>
3	CH0COMSEN	<p>Channel 0 compare output shadow enable</p> <p>When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled.</p> <p>0: Channel 0 output compare shadow disable</p> <p>1: Channel 0 output compare shadow enable</p> <p>The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 00.</p>
2	CH0COMFEN	<p>Channel 0 output compare fast enable</p> <p>When this bit is set, the effect of an event on the trigger input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison.</p> <p>0: Channel 0 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles.</p> <p>1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.</p>
1:0	CH0MS[1:0]	<p>Channel 0 I/O mode selection</p> <p>This bit-field specifies the work mode of the channel and the input signal selection.</p> <p>This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).</p> <p>00: Channel 0 is configured as output</p> <p>01: Channel 0 is configured as input, IS0 is connected to CI0FE0</p> <p>10: Channel 0 is configured as input, IS0 is connected to CI1FE0</p> <p>11: Channel 0 is configured as input, IS0 is connected to ITS. This mode is working</p>

only if an internal trigger input is selected through TRGS bits in TIMERx_SMCFG register.

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 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 TIMERx_CHCTL2 register is clear. 00: Prescaler disable, 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 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

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 CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH0COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMSEN description
9:8	CH3MS[1:0]	Channel 3 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. (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 TRGS bits in TIMERx_SMCFG 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

		ETIF input. 0: Channel 2 output compare clear disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	<p>Channel 2 compare output control</p> <p>This bit-field controls the behavior of the output reference signal O2CPRE which drives CH2_O and CH2_ON. O2CPRE is active high, while CH2_O and CH2_ON active level depends on CH2P and CH2NP bits.</p> <p>000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT.</p> <p>001: Set the channel output. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV.</p> <p>010: Clear the channel output. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV.</p> <p>011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV.</p> <p>100: Force low. O2CPRE is forced low level.</p> <p>101: Force high. O2CPRE is forced high level.</p> <p>110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active.</p> <p>111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive.</p> <p>When configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “Timing mode” mode to “PWM” mode or when the result of the comparison changes.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 00(COMPARE MODE).</p>
3	CH2COMSEN	<p>Channel 2 compare output shadow enable</p> <p>When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled.</p> <p>0: Channel 2 output compare shadow disable 1: Channel 2 output compare shadow enable</p> <p>The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 00.</p>
2	CH2COMFEN	<p>Channel 2 output compare fast enable</p> <p>When this bit is set, the effect of an event on the trigger input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a</p>

compare match, and CH2_O is set to the compare level independently from the result of the comparison.

0: Channel 2 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH2_O output is 5 clock cycles.

1: Channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.

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. This bit-field is writable only when the channel is not active. (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 TRGS bits in TIMERx_SMCFG 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 CH0CAPFLT description
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH0CAPPSC 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 disable, fSAMP=fDTS, N=1 0001: fSAMP=fCK_TIMER, N=2 0010: fSAMP=fCK_TIMER, N=4 0011: fSAMP=fCK_TIMER, N=8 0100: fSAMP=fDTS/2, N=6 0101: fSAMP=fDTS/2, N=8 0110: fSAMP=fDTS/4, N=6 0111: fSAMP=fDTS/4, N=8 1000: fSAMP=fDTS/8, N=6 1001: fSAMP=fDTS/8, N=8 1010: fSAMP=fDTS/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 TIMERx_CHCTL2 register is clear. 00: Prescaler disable, 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 (TIMERx_CHCTL2)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
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	CH3NP	Channel 3 complementary output 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 output polarity Refer to CH0NP description
10	CH2NEN	Channel 2 complementary output enable

		Refer to CH0NEN description
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 output polarity Refer to CH0NP description
6	CH1NEN	Channel 1 complementary output enable Refer to CH0NEN description
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 output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, In conjunction with CH0P, this bit is used to define the polarity of Cl0. This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
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 Cl0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for Cl0FE0 or Cl1FE0. [CH0NP==0, CH0P==0]: ClxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will not be inverted. [CH0NP==0, CH0P==1]: ClxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will be inverted.

[CH0NP==1, CH0P==0]: Reserved.

[CH0NP==1, CH0P==1]: ClxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And ClxFE0 will be not inverted.

This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.

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).

31	30	29	28	27	26	25	24	23	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-filed indicates the current counter value. Writing to this bit-filed 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).

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	<p>Prescaler value of the counter clock</p> <p>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.</p>

Counter auto reload register (TIMERx_CAR)

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	<p>Counter auto reload value</p> <p>This bit-field specifies the auto reload value of the counter.</p>

Counter repetition register (TIMERx_CREP)

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

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.

7:0	CREP[7:0]	Counter repetition value This bit-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

Bits	Fields	Descriptions
------	--------	--------------

31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	<p>Capture or compare value of channel1</p> <p>When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only.</p> <p>When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.</p>

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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	<p>Capture or compare value of channel 2</p> <p>When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only.</p> <p>When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.</p>

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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

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															
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					rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	POEN	Primary output enable This bit is set by software or automatically 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 CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN 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 can be not set by hardware. 1: POEN can be set by hardware automatically at the next update event, 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 polarity

		This bit specifies the polarity of the BRKIN input signal. 0: BRKIN input active low 1: BRKIN input active high
12	BRKEN	<p>Break enable</p> <p>This bit can be set to enable the BRKIN and CCS clock failure event inputs.</p> <p>0: Break inputs disabled 1: Break inputs enabled</p> <p>This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.</p>
11	ROS	<p>Run mode off-state configure</p> <p>When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode.</p> <p>0: When POEN bit is set, the channel output signals (CHx_O/CHx_ON) are disabled. 1: When POEN bit is set, the channel output signals (CHx_O/CHx_ON) are enabled, with relationship to CHxEN/CHxNEN bits in TIMERx_CHCTL2 register.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.</p>
10	IOS	<p>Idle mode off-state configure</p> <p>When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode.</p> <p>0: When POEN bit is reset, the channel output signals (CHx_O/CHx_ON) are disabled. 1: When POEN bit is reset, the channel output signals (CHx_O/CHx_ON) are enabled, with relationship to CHxEN/CHxNEN bits in TIMERx_CHCTL2 register.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.</p>
9:8	PROT[1:0]	<p>Complementary register protect control</p> <p>This bit-field specifies the write protection property of registers.</p> <p>00: protect disable. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected. 11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/CHxCOMSEN bits in TIMERx_CHCTL0/1 registers (if the related channel is configured in output) are writing protected.</p> <p>This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.</p>
7:0	DTCFG[7:0]	Dead time configure

This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow :

DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}.

DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0])x t_{DT}, t_{DT}=t_{DTS}*2.

DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0])x t_{DT}, t_{DT}=t_{DTS}*8.

DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0])x 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.

DMA configuration register (TIMERx_DMACFG)

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

rw

rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMA TB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed define the first address for the DMA access the TIMERx_DMA TB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMA TB, you will access the address of start address + 0x4. 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... In a word: Start Address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

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

rw

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 range (Start Addr + Transfer Timer * 4) will be accessed. The transfer Timer 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).

31	30	29	28	27	26	25	24	23	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 OUTSEL

rw rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 0: No effect 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored
0	OUTSEL	The output value selection This bit-field set and reset by software. 0: No effect 1: If POEN and IOS is 0, the output disabled

17.2. General level0 timer(TIMERx, x=1, 2, 3, 4)

17.2.1. Overview

The general level0 timer module (Timer1, 2, 3, 4) is a four-channel timer that supports input capture, output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 time reference is a 16-bit(TIMER3/4) counter or 32-bit(TIMER1/2) that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used to count or time external events that drive other timers.

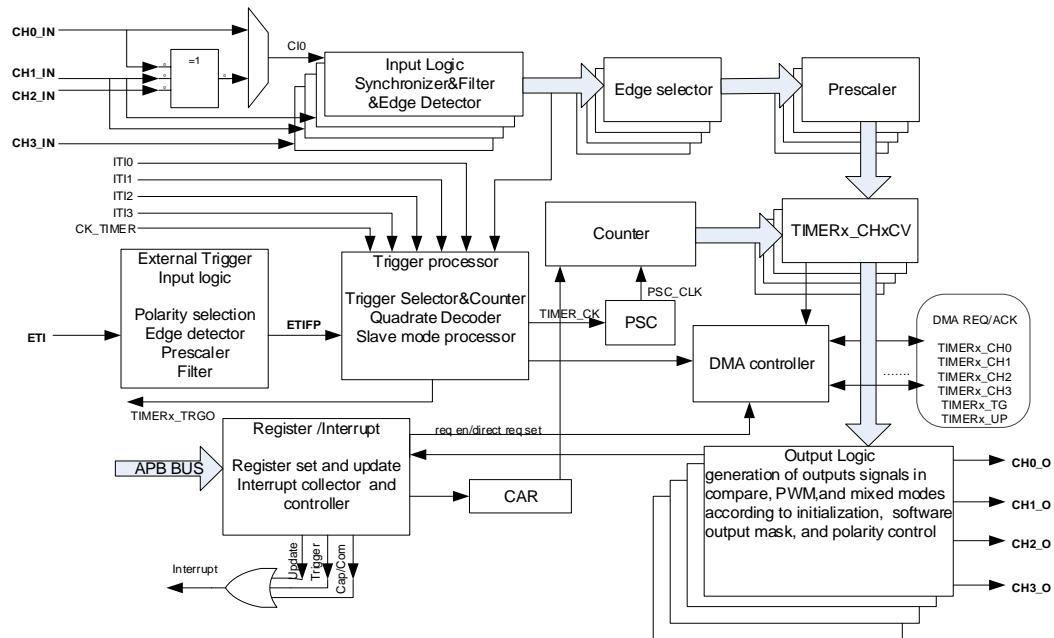
Timer and timer are completely independent, but there may be synchronized to provide a larger timer with their counters incrementing in unison.

17.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16bit(TIMER3/4),32bit(TIMER1/2).
- Source of count clock is selectable:
internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: count up, count down, count up/down.
- Quadrature decoder: used to track motion and determine both rotation direction and position.
- Hall sensor: for 3-phase motor control.
- Programmable prescaler: 16 bit. Factor can be changed on the go.
- Each channel is user-configurable:
Input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Auto-reload function.
- Interrupt output or DMA request on: update, trigger event, and compare/capture event.
- Daisy chaining of timer modules to allow a single timer to initiate multiple timing events.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer Master/Slave mode controller.

17.2.3. Block diagram

[Figure 17-34. General Level 0 timer block diagram](#) provides details on the internal configuration of the general level0 timer.

Figure 17-34. General Level 0 timer block diagram


17.2.4. Function overview

Clock selection

The general level0 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by SMC (TIMERx_SMCFG bit [2:0]).

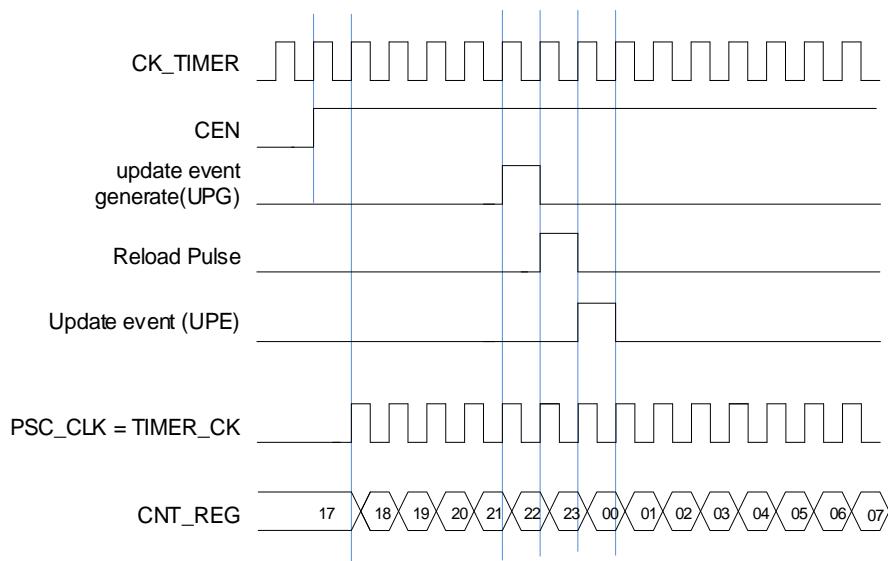
- SMC [2:0] == 3'b000. Internal timer clock CK_TIMER which is from module RCU.

The default internal clock source is the CK_TIMER used to drive the counter prescaler when the slave mode is disabled (SMC [2:0] == 3'b000). When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU.

If the slave mode controller is enabled by setting SMC [2:0] in the TIMERx_SMCFG register to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRGS [2:0] in the TIMERx_SMCFG register and described as follows. When the slave mode selection bits SMC [2:0] are set to 0x4, 0x5 or 0x6, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 17-35. Normal mode, internal clock divided by 1



■ SMC [2:0] == 3'b111(external clock mode 0). External input pin 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 TIMERx_CI0/TIMERx_CI1. This mode can be selected by setting SMC [2:0] to 0x7 and the TRGS [2:0] to 0x4, 0x5 or 0x6.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting SMC [2:0] to 0x7 and the TRGS [2:0] to 0x0, 0x1, 0x2 or 0x3.

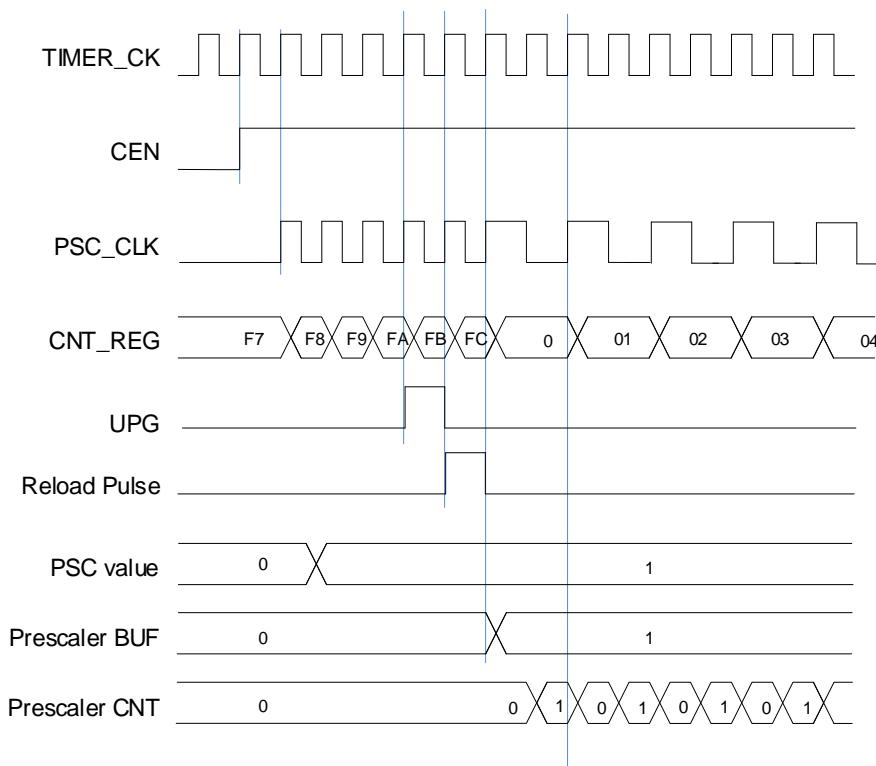
■ SMC1== 1'b1(external clock mode 1). External input pin source (ETI)

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 TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is to set the SMC [2:0] to 0x7 and the TRGS [2:0] to 0x7 respectively. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as 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.

Prescaler

The prescaler can divide the timer clock (TIMER_CK) to the counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled through prescaler register (TIMERx_PSC) which can be changed on the go but be taken into account at the next update event.

Figure 17-36. 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 to count once again from 0. The update event is generated at each counter overflow. The counting direction bit `DIR` in the `TIMERx_CTL1` register should be set to 0 for the up counting mode.

When the update event is set by the `UPG` bit in the `TIMERx_SWEVG` register, the counter value will be initialized to 0 and generates an update event.

If the `UPDIS` bit in `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

[Figure 17-37. Timing chart of up counting mode, PSC=0/1](#) and [Figure 17-38. Timing chart of up counting mode, change TIMERx_CAR ongoing](#) show some examples of the counter behavior for different clock prescaler factor when `TIMERx_CAR`=0x63.

Figure 17-37. Timing chart of up counting mode, PSC=0/1

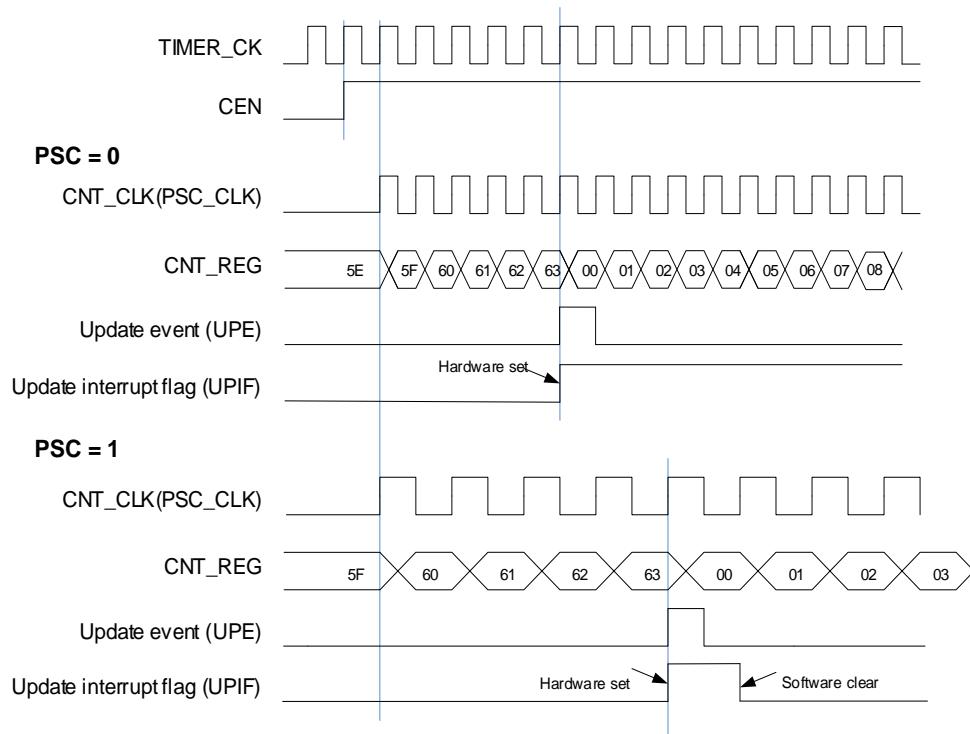
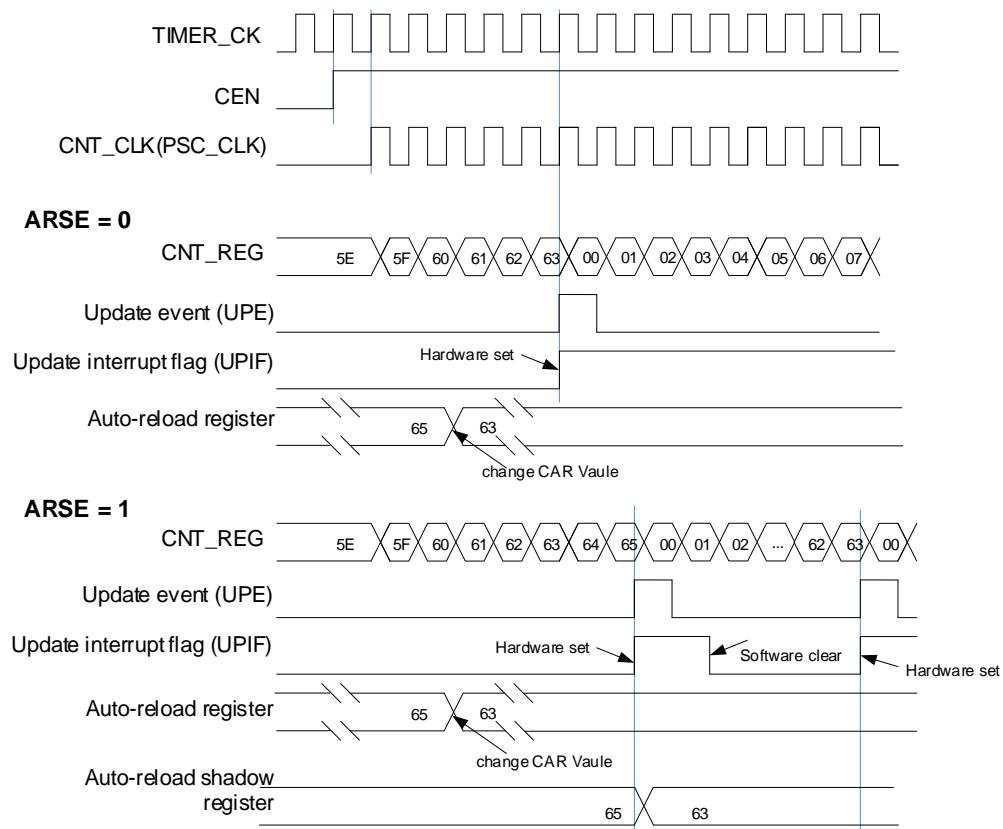


Figure 17-38. Timing chart of up counting mode, change TIMERx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter-reload value, which is defined in the `TIMERx_CAR` register, to 0 in a count-down direction. Once the counter reaches to 0, the counter restarts to count again from the counter-reload value. If the repetition counter is set, the update event was generated after the number (`TIMERx_CREP+1`) of underflow. Else the update event is generated at each counter underflow. 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 generates an update event.

If the `UPDIS` bit in `TIMERx_CTL0` register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

[Figure 17-39. Timing chart of down counting mode,PSC=0/1](#) and [Figure 17-40. Timing chart of down counting mode, change TIMERx CAR ongoing](#) show some examples of the counter behavior for different clock frequencies when `TIMERx_CAR`=0x63.

Figure 17-39. Timing chart of down counting mode,PSC=0/1

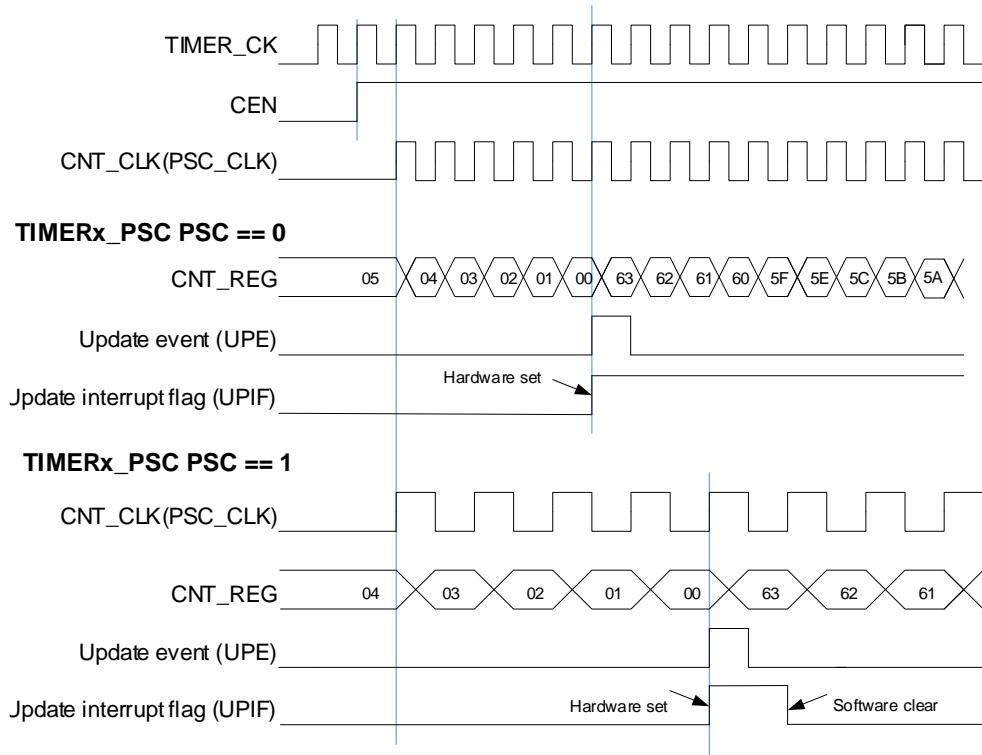
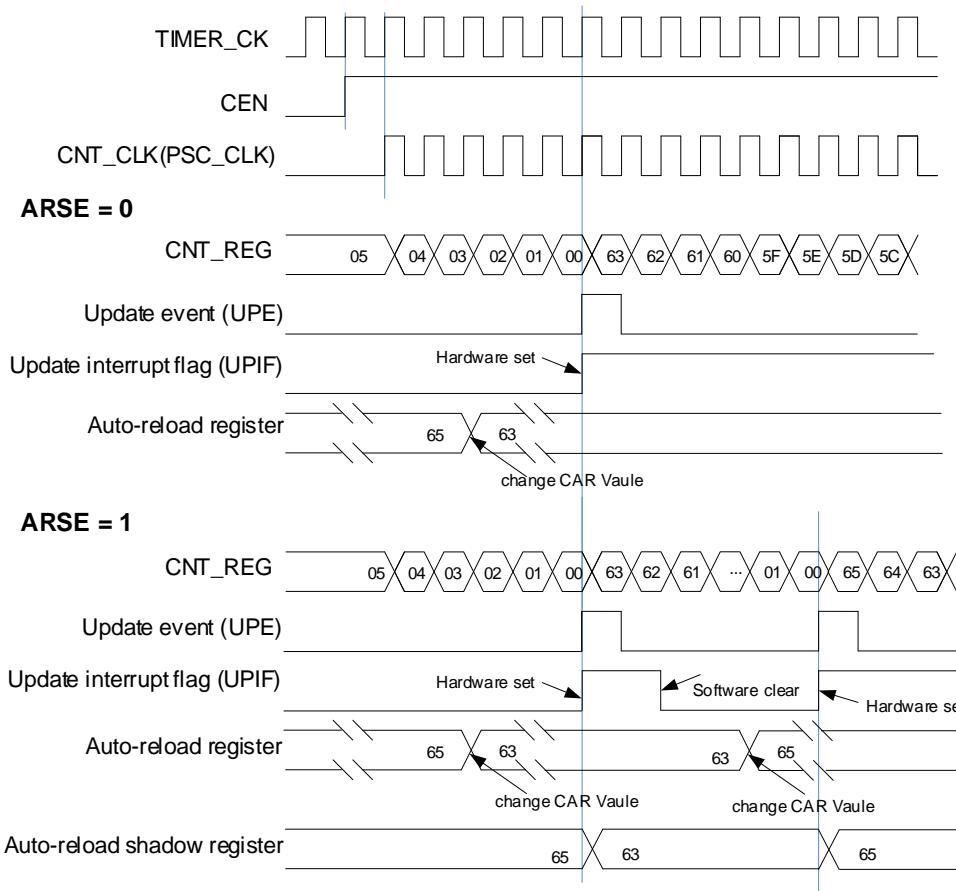


Figure 17-40. Timing chart of down counting mode, change TIMERx_CAR ongoing



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 the counter-reload value subtract 1 in the up-counting direction and generates an underflow event when the counter counts to 1 in the down-counting mode. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned 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 generates an update event irrespective of whether the counter is counting up or down in the center-align counting mode.

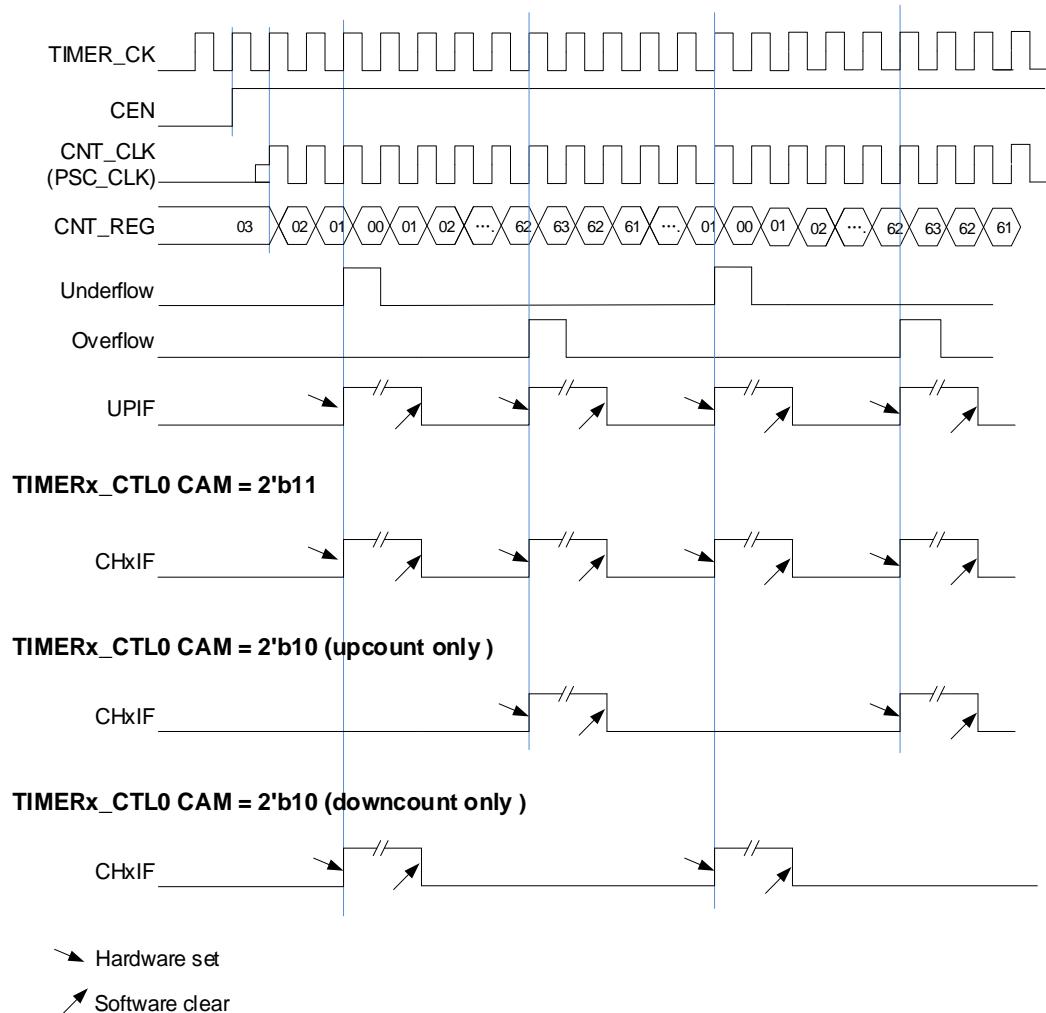
The UPIF bit in the TIMERx_INTF register can be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to [Figure 17-41. Timing chart of center-aligned counting mode](#).

If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

Figure 17-41. Timing chart of center-aligned counting mode show some examples of the counter behavior when TIMERx_CAR=0x63. TIMERx_PSC=0x0

Figure 17-41. Timing chart of center-aligned counting mode



Capture/compare channels

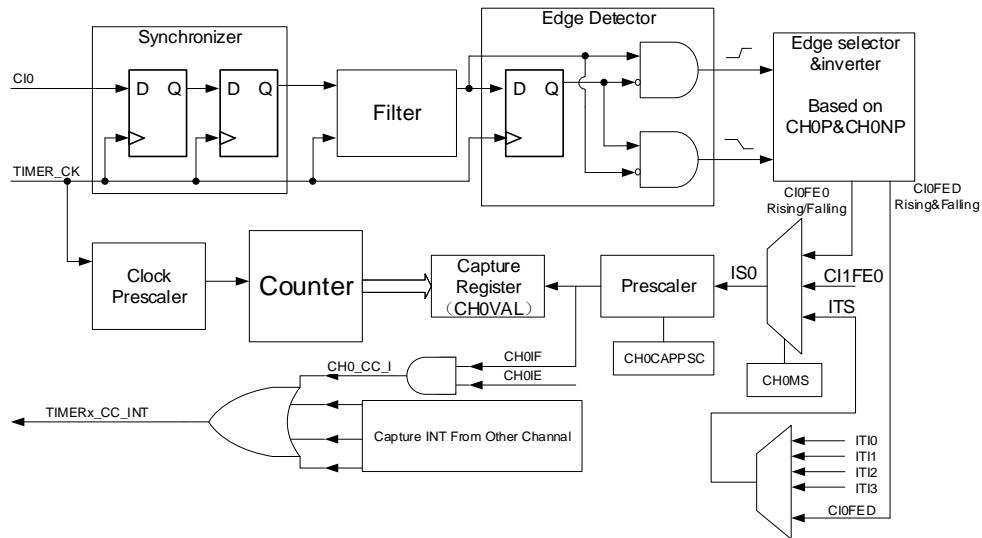
The general level0 Timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

- Input capture mode

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 enabled by CHxIE = 1.

Figure 17-42. Input capture logic



One of channels' input signals (CIx) can be chosen from the TIMERx_CHx signal or the Exclusive-OR function of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the channel input signal (CIx) is synchronized to TIMER_CK 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 by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, TIMERx_CHxCV will restore the value of Counter.

So the process can be divided to several steps as below:

Step1: Filter Configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge Selection. (CHxP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP.

Step3: Capture source Selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel to input mode (CHxMS!=0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable. (CHxIE and CHxDEN in TIMERx_DMAINTEN)

Enable the related interrupt enable; you can got the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: When you wanted input signal is got, TIMERx_CHxCV will be set by counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt

and DMA request will be asserted based on the your configuration of CHxIE and CHxDEN in TIMERx_DMINTEN

Direct generation: If you want to generate a DMA request or interrupt, you can set CHxG by software directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

- Output compare mode

Figure 17-43. Output compare logic (x=0,1,2,3)

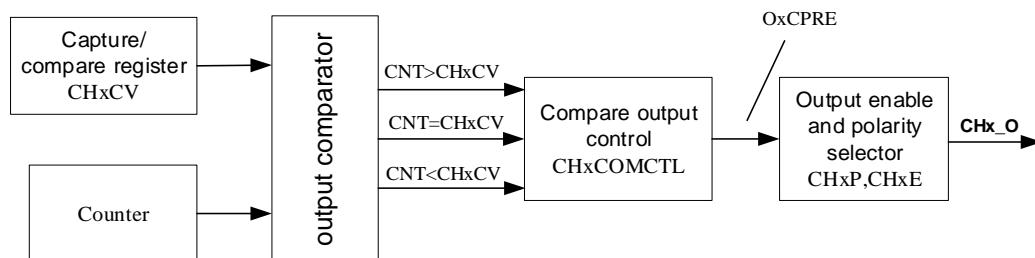


Figure 17-43. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to [Channel output reference signal](#)) is described as below: The active level of O0CPRE is high, the output level of CH0_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), CHxE=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 CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CxCDE=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 shadow enable mode by CHxCOMSEN
- * Set the output mode (Set/Clear/Toggle) by CHxCOMCTL.
- * Select the active high polarity by CHxP
- * Enable the output by CHxEN

Step3: Interrupt/DMA-request enables configuration by CHxIE/CxCDE

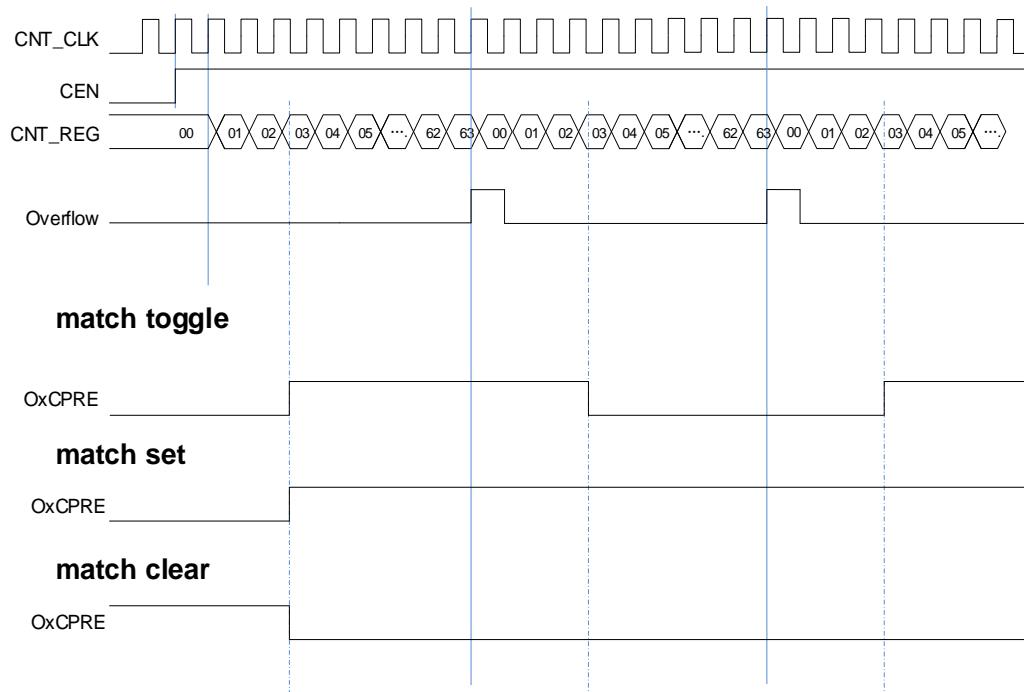
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

About the CHxVAL, you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

[**Figure 17-44. Output-compare in three modes**](#) show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 17-44. Output-compare in three modes



PWM mode

In the output PWM mode (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b111(PWM mode1), the channel can outputs PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, we have can also divide PWM into EAPWM (Edge aligned PWM) and CAPWM (Centre aligned PWM).

The EAPWM period is determined by TIMERx_CAR and duty cycle is by TIMERx_CHxCV.

[**Figure 17-45. 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 17-46. Timing chart of CAPWM](#) shows the CAPWM output and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR , the output will be always active under PWM mode0 ($\text{CHxCOMCTL} == 3'b110$).

And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 ($\text{CHxCOMCTL} == 3'b110$).

Figure 17-45. Timing chart of EAPWM

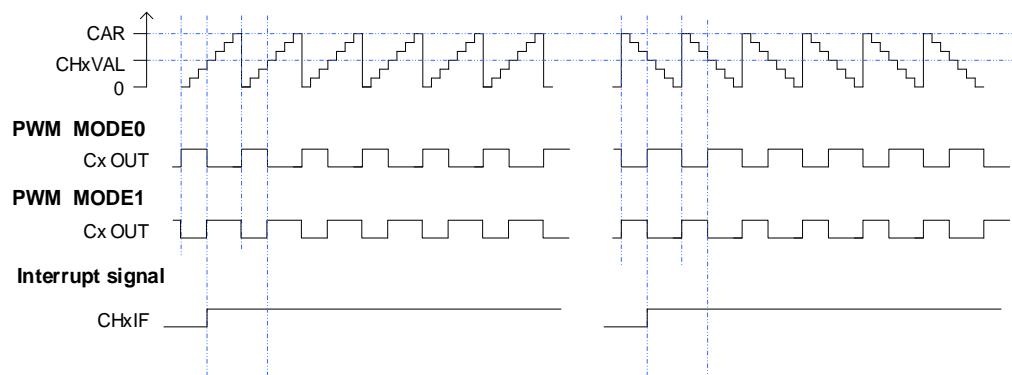
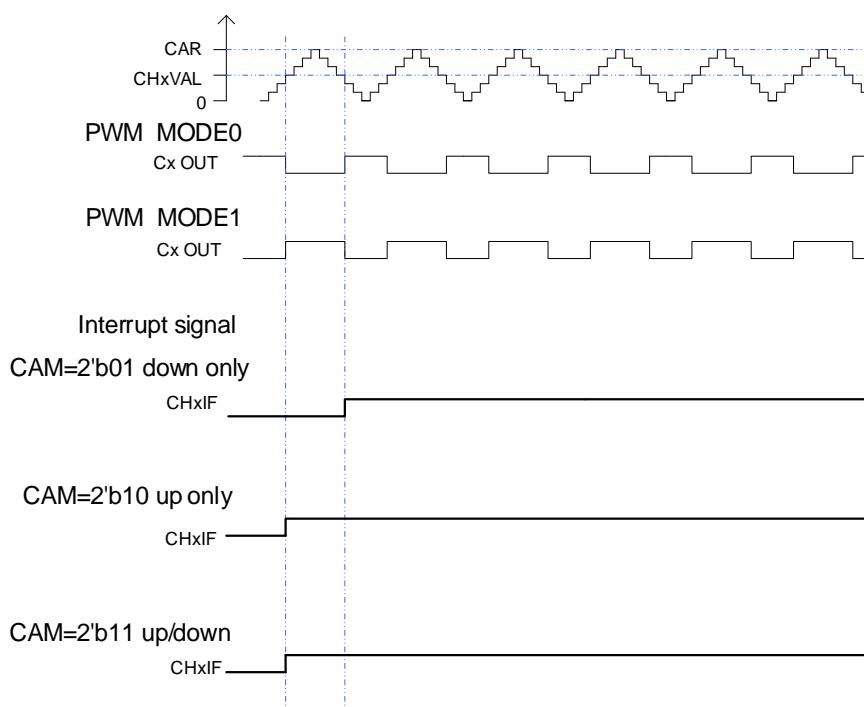


Figure 17-46. Timing chart of CAPWM



Channel output reference signal

As is shown in [Figure 17-43. Output compare logic \(x=0,1,2,3\)](#), when the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL field. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

The OxCPRE signal can be forced to 0 when the ETIFE signal is derived from the external ETI pin and when it is set to a high level by setting the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register. The OxCPRE signal will not return to its active level until the next update event occurs.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting SMC=0x01, 0x02, or 0x03 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 17-5. Counting direction versus encoder signals](#). 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 TIMERx_CAR register before the counter starts to count.

Table 17-5. Counting direction versus encoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 only counting	CI1FE1=High	Down	Up	-	-
	CI1FE1=Low	Up	Down	-	-
CI1 only	CI0FE0=High	-	-	Up	Down

counting	Cl0FE0=Low	-	-	Dow n	Up
Cl0 and Cl1 counting	Cl1FE1=High	Dow n	Up	X	X
	Cl1FE1=Low	Up	Dow n	X	X
	Cl0FE0=High	X	X	Up	Dow n
	Cl0FE0=Low	X	X	Dow n	Up

Note: "-" means "no counting"; "X" means impossible.

Figure 17-47. Example of counter operation in encoder interface mode

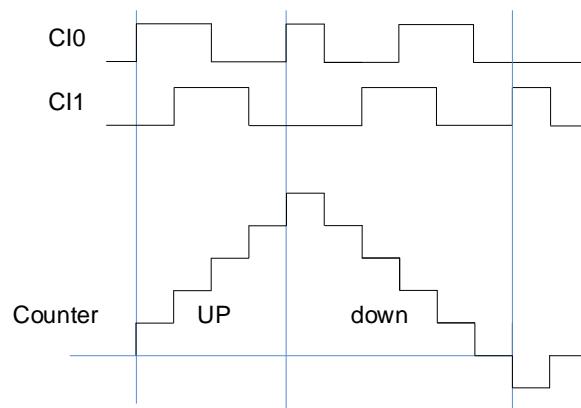
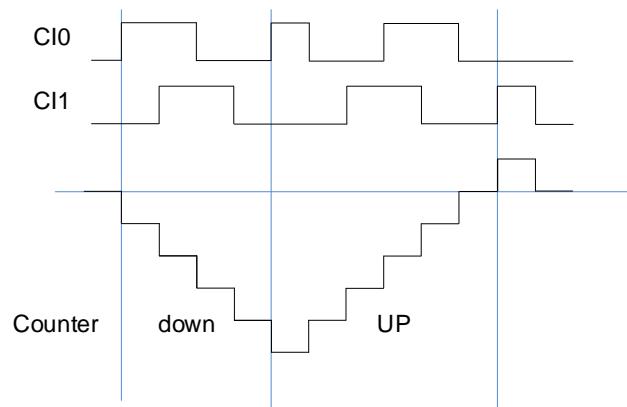


Figure 17-48. Example of encoder interface mode with Cl0FE0 polarity inverted



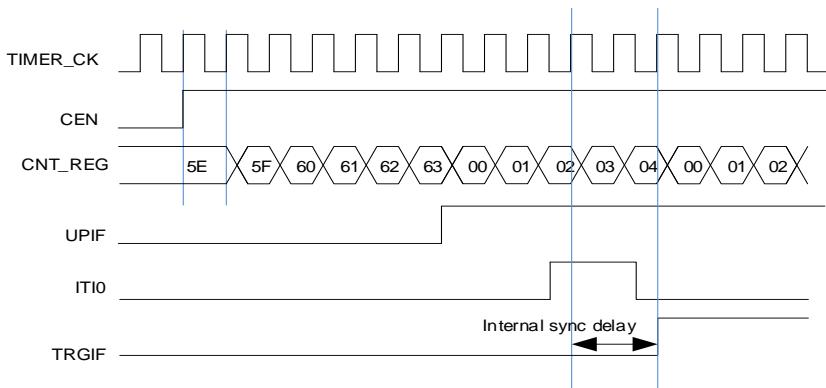
Hall sensor function

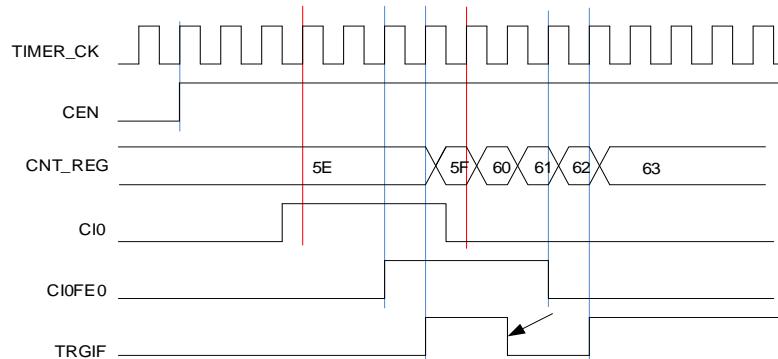
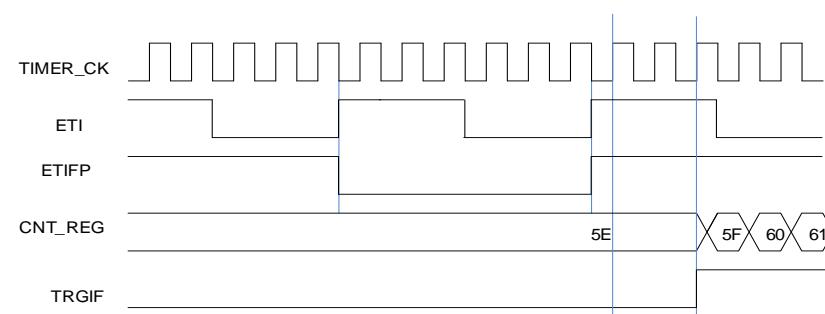
Refer to [Advanced timer \(TIMERx, x=0\)](#).

Slave controller

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode which is selected by the SMC [2:0] in the TIMERx_SMCFG register. The trigger input of these modes can be selected by the TRGS [2:0] in the TIMERx_SMCFG register.

Table 17-6. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP 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, 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.
	Restart mode The counter will be cleared and restart when a rising edge of trigger input comes.	TRGS[2:0] = 3'b000 ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
Exam1	Figure 17-49. 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.	TRGS[2:0]=3'b101 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 17-50. Pause mode			
		 <p>The diagram shows the waveforms for Pause mode. The TIMER_CK signal is a square wave. The CEN signal is high during the first four cycles of TIMER_CK. The CNT_REG counter starts at 5E and counts up to 63. The CI0 and CI0FE0 compare values are shown as step functions. The TRGIF flag is set to 1 during the fifth cycle of TIMER_CK.</p>		
	Event mode The counter will start to count when a rising edge of trigger input comes.	TRGS[2:0] = 3'b111 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.
Exam3	Figure 17-51. Event mode			
		 <p>The diagram shows the waveforms for Event mode. The TIMER_CK signal is a square wave. The ETI signal has a single rising edge. The ETIFP signal is high during the rising edge of ETI. The CNT_REG counter starts at 5E and counts up to 61. The TRGIF flag is set to 1 during the rising edge of ETI.</p>		

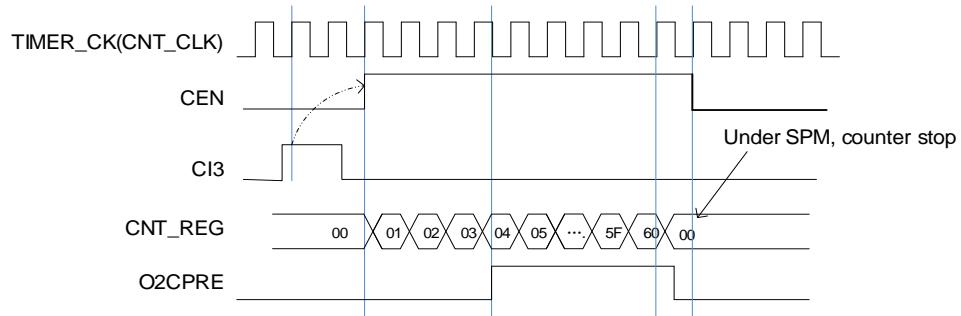
Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in **TIMERx_CTL0**. When you set SPM, the counter will be clear and stop when the next update event automatically. In order to get pulse waveform, you can set the **TIMERx** to PWM mode or compare by **CHxCOMCTL**.

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit **CEN** in the **TIMERx_CTL0** register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the **CEN** bit to 1 using software. Setting the **CEN** bit to 1 or a trigger from the trigger signals 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 using software, the counter will be stopped and its value 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 trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0/1 register. After a trigger rising occurs in the single pulse mode, the OxCMPRE signal will immediately be forced to the state which the OxCMPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 17-52. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Refer to [Advanced timer \(TIMERx, x=0\)](#).

Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB; Of course, you have to enable a DMA request which will be asserted by some internal interrupt event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB 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), then the timer's DMA request is finished. 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 DMASAR+0x4, DMASAR+0x8, DMASAR+0xc at the next 3 accesses to TIMERx_DMATB. In one word, one time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex™-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

17.2.5. **TIMERx registers(x=1, 2, 3, 4)**

TIMER1 secure access base address: 0x5000 0000
 TIMER1 non-secure access base address: 0x4000 0000
 TIMER2 secure access base address: 0x5000 0400
 TIMER2 non-secure access base address: 0x4000 0400
 TIMER3 secure access base address: 0x5000 0800
 TIMER3 non-secure access base address: 0x4000 0800
 TIMER4 secure access base address: 0x5000 0C00
 TIMER4 non-secure access base address: 0x4000 0C00

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																				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved					CKDIV[1:0]		ARSE		CAM[1:0]		DIR		SPM		UPS		UPDIS		CEN	
					rw		rw		rw		rw		rw		rw		rw			

Bits	Fields	Descriptions
31:10	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 the timer clock (TIMER_CK) and the dead-time and sampling clock (DTS), which is used by the dead-time generators and the digital filters. 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 TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter aligns 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 under center-aligned and channel is configured in output mode (CHxMS=00 in

`TIMERx_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 under center-aligned and channel is configured in output mode (`CHxMS=00` 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 under center-aligned and channel is configured in output mode (`CHxMS=00` 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, cannot be switched from 0x00 to non 0x00.

4	<code>DIR</code>	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.
3	<code>SPM</code>	Single pulse mode. 0: Counter continues after update event. 1: The CEN is cleared by hardware and the counter stops at next update event.
2	<code>UPS</code>	Update source This bit is used to select the update event sources by software. 0: When enabled, any of the following events generate an update interrupt or DMA request: The UPG bit is set The counter generates an overflow or underflow event The slave mode controller generates an update event. 1: When enabled, only counter overflow/underflow generates an update interrupt or DMA request.
1	<code>UPDIS</code>	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 if the slave mode controller generates a hardware reset event.
0	<code>CEN</code>	Counter enable 0: Counter disable 1: Counter enable

The CEN bit must be set by software when timer works in external clock, pause mode and encoder 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).

31	30	29	28	27	26	25	24	23	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								TIOS	MMC[2:0]	DMAS	Reserved				
rw								rw	rw	rw					

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TIOS	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, CH1 and 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 in master mode to slave timers 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 useful to start several timers at the same time or to control a window in which a slave timer is enabled. In this mode the master mode controller selects the counter enable signal TIMERx_EN 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 occurred. 100: Compare. In this mode the master mode controller selects the O0CPRE signal is used as TRGO 101: Compare. In this mode the master mode controller selects the O1CPRE signal

is used as TRGO

110: Compare. In this mode the master mode controller selects the O2CPRE signal is used as TRGO

111: Compare. In this mode the master mode controller selects the O3CPRE signal is used as TRGO

3	DMAS	DMA request source selection 0: DMA request of channel x is sent when channel x event occurs. 1: DMA request of channel x 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		TRGS[2:0]	Reserved		SMC[2:0]				

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 SMC for enable External clock mode1. In external clock mode 1, the counter is clocked by any active edge on the ETIF 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 TRGS bits must not be 3'b111 in this case. The external clock input will be ETIF if external clock mode 0 and external clock mode 1 are enabled at the same time. Note: External clock mode 0 enable is in this register's SMC bit-field.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETI must not be at higher than 1/4 of

TIMER_CK frequency. When the external trigger signal is a fast clock, the prescaler can be enabled to reduce ETI frequency.

- 00: Prescaler disable
- 01: ETI frequency will be divided by 2
- 10: ETI frequency will be divided by 4
- 11: ETI frequency will be 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 ETI signal and the length of the digital filter applied to ETI.
		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 selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected together.
		0: Master-slave mode disable
		1: Master-slave mode enable

6:4	TRGS[2:0]	Trigger selection
		This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter.
		000: Internal trigger input 0 (ITI0)
		001: Internal trigger input 1 (ITI1)
		010: Internal trigger input 2 (ITI2)
		011: Internal trigger input 3 (ITI3)
		100: Cl0 edge flag (Cl0F_ED)
		101: channel 0 input Filtered output (Cl0FE0)
		110: channel 1 input Filtered output (Cl1FE1)

111: External trigger input filter output(ETIFP)

These bits must not be changed when slave mode is enabled.

3	Reserved	Must be kept at reset value.
2:0	SMC[2:0]	Slave mode control
		000: Disable mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high.
		001: Quadrature decoder mode 0.The counter counts on Cl1FE1 edge, while the direction depends on Cl0FE0 level.
		010: Quadrature decoder mode 1.The counter counts on Cl0FE0 edge, while the direction depends on Cl1FE1 level.
		011: Quadrature decoder mode 2.The counter counts on both Cl0FE0 and Cl1FE1 edge, while the direction depends on each other.
		100: Restart mode. The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input.
		101: Pause mode. The trigger input enables the counter clock when it is high and disables the counter when it is low.
		110: Event mode. A rising edge of the trigger input enables the counter. The counter cannot be disabled by the slave mode controller.
		111: External clock mode0. The counter counts on the rising edges of the selected trigger.

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	Reserved	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	Reserved	TRGIE	Reserved	CH3IE	CH2IE	CH1IE	CHOIE	UPIE
	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	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).

31	30	29	28	27	26	25	24	23	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	CH3OF	CH2OF	CH1OF	CH0OF	Reserved	TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF	rc_w0	rc_w0	rc_w0

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 on trigger input generates a trigger event. When the slave mode controller is enabled in pause mode both edges on trigger input generates a trigger event. 0: No trigger event occurred. 1: Trigger interrupt occurred.
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3's capture/compare interrupt enable Refer to CH0IF description
3	CH2IF	Channel 2's capture/compare interrupt enable Refer to CH0IF description
2	CH1IF	Channel 1's capture/compare interrupt flag

Refer to CH0IF description

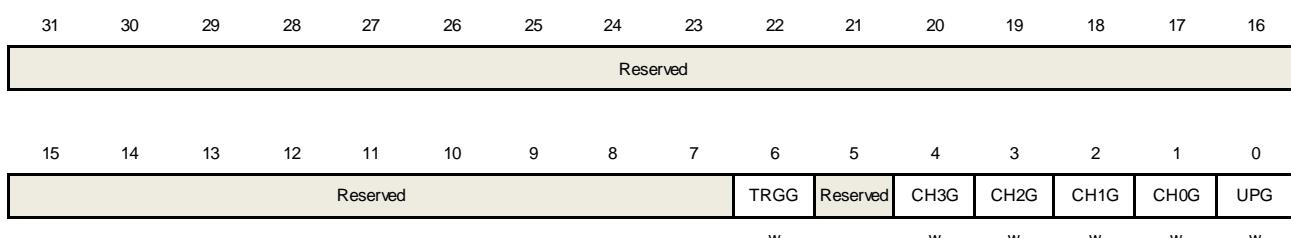
1	CH0IF	Channel 0's capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. If Channel0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No Channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

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	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_STAT register is 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's capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2's capture or compare event generation Refer to CH0G description

2	CH1G	Channel 1's capture or compare event generation Refer to CH0G description
1	CH0G	<p>Channel 0's capture or compare event generation</p> <p>This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high.</p> <p>0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event</p>
0	UPG	<p>This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time.</p> <p>0: No generate an update event 1: Generate an update event</p>

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]	CH1COMSEN	CH1COMFEN	CH1MS[1:0]	CH0COMCEN	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]	CH0CAPFLT[3:0]	CH0CAPPSC[1:0]												
rw		rw		rw		rw		rw		rw		rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH1COMCEN	<p>Channel 1 output compare clear enable</p> <p>Refer to CH0COMCEN description</p>
14:12	CH1COMCTL[2:0]	<p>Channel 1 compare output control</p> <p>Refer to CH0COMCTL description</p>
11	CH1COMSEN	Channel 1 output compare shadow enable

		Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMSEN description
9:8	CH1MS[1:0]	Channel 1 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. (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 CI0FE1 10: Channel 1 is configured as input, IS1 is connected to CI1FE1 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 TRGS bits in TIMERx_SMCFG 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 ETIF input. 0: Channel 0 output compare clear disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of the output reference signal O0CPRE which drives CH0_O. O0CPRE is active high, while CH0_O active level depends on CH0P bits. 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. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced low level. 101: Force high. O0CPRE is forced high level. 110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active. 111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive. When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from "Timing mode" mode to "PWM" mode or when the result of the comparison changes.

3	CH0COMSEN	<p>Channel 0 compare output shadow enable</p> <p>When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled.</p> <p>0: Channel 0 output compare shadow disable</p> <p>1: Channel 0 output compare shadow enable</p> <p>The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).</p>
2	CH0COMFEN	<p>Channel 0 output compare fast enable</p> <p>When this bit is set, the effect of an event on the trigger input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison.</p> <p>0: Channel 0 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles.</p> <p>1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.</p>
1:0	CH0MS[1:0]	<p>Channel 0 I/O mode selection</p> <p>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. (CH0EN bit in TIMERx_CHCTL2 register is reset).).</p> <p>00: Channel 0 is configured as output</p> <p>01: Channel 0 is configured as input, IS0 is connected to CI0FE0</p> <p>10: Channel 0 is configured as input, IS0 is connected to CI1FE0</p> <p>11: Channel 0 is configured as input, IS0 is connected to ITS. This mode is working only if an internal trigger input is selected through TRGS bits in TIMERx_SMCFG register.</p>

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	<p>Channel 1 input capture filter control</p> <p>Refer to CH0CAPFLT description</p>
11:10	CH1CAPPSC[1:0]	<p>Channel 1 input capture prescaler</p> <p>Refer to CH0CAPPSC description</p>
9:8	CH1MS[1:0]	<p>Channel 1 mode selection</p> <p>Same as Output compare mode</p>
7:4	CH0CAPFLT[3:0]	<p>Channel 0 input capture filter control</p> <p>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</p>

signal and the length of the digital filter applied to C10.

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 clear.
	00:	Prescaler disable, 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 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		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 CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH0COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMSEN description
9:8	CH3MS[1:0]	Channel 3 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. (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 CI2FE3 10: Channel 3 is configured as input, IS3 is connected to CI3FE3 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 TRGS bits in TIMERx_SMCFG 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 ETIF input. 0: Channel 2 output compare clear disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of the output reference signal O2CPRE which drives CH2_O. O2CPRE is active high, while CH2_O and active level depends on CH2P bits. 000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 001: Set the channel output. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 010: Clear the channel output. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 100: Force low. O2CPRE is forced low level. 101: Force high. O2CPRE is forced high level.

110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active.

111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive.

When configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “Timing mode” mode to “PWM” mode or when the result of the comparison changes.

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 disable 1: Channel 2 output compare shadow enable 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 When this bit is set, the effect of an event on the trigger input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
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. This bit-field is writable only when the channel is not active. (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 TRGS bits in TIMERx_SMCFG 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 CH0CAPFLT description
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH0CAPPSC 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 disable, $f_{SAMP}=f_{DTS}$, $N=1$ 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 TIMERx_CHCTL2 register is clear. 00: Prescaler disable, 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 (TIMERx_CHCTL2)

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															

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	CH3NP	Channel 3 complementary output 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 output 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 output 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 output polarity When channel 0 is configured in output mode, this bit should be keep reset value. When channel 0 is configured in input mode, In conjunction with CH0P, this bit is used to define the polarity of Cl0.
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 CI0 signal polarity.

[CH0NP, CH0P] will select the active trigger or capture polarity for Cl0FE0 or Cl1FE0.

[CH0NP==0, CH0P==0]: ClxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will not be inverted.

[CH0NP==0, CH0P==1]: ClxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will be inverted.

[CH0NP==1, CH0P==0]: Reserved.

[CH0NP==1, CH0P==1]: ClxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And ClxFE0 will be not inverted.

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) (x=1,2)

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
CNT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

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.

Counter register (TIMERx_CNT) (x=3,4)

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															
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 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).

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

rw

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) (x=1,2)

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
CARL[31:16]															

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

Bits	Fields	Descriptions
31:0	CARL[31:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter auto reload register (TIMERx_CAR) (x=3,4)

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

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) (x=1,2)

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
CH0VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Fields	Descriptions
------	--------	--------------

31:0	CH0VAL[31:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.
------	--------------	--

Channel 0 capture/compare value register (TIMERx_CH0CV) (x=3,4)

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV) (x=1,2)

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
CH1VAL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

Bits	Fields	Descriptions
------	--------	--------------

31:0	CH1VAL[31:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.
------	--------------	--

Channel 1 capture/compare value register (TIMERx_CH1CV) (x=3,4)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV) (x=1,2)

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
CH2VAL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
------	--------	--------------

31:0	CH2VAL[31:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.
------	--------------	---

Channel 2 capture/compare value register (TIMERx_CH2CV) (x=3,4)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV) (x=1,2)

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
CH3VAL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
------	--------	--------------

31:0	CH3VAL[31:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.
------	--------------	--

Channel 3 capture/compare value register (TIMERx_CH3CV) (x=3,4)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

DMA configuration register (TIMERx_DMACFG)

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

Bits	Fields	Descriptions
------	--------	--------------

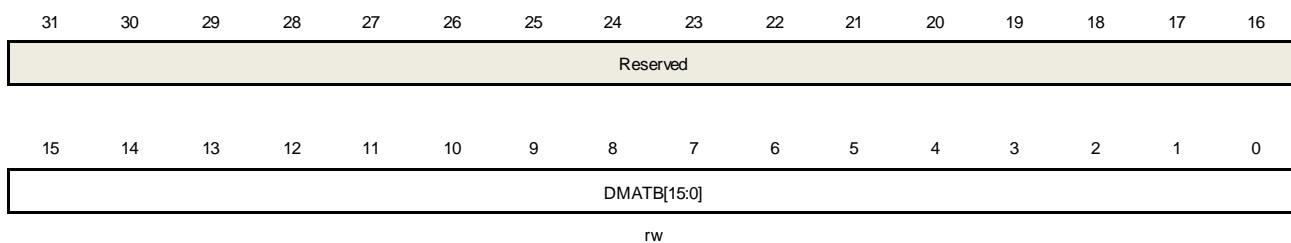
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMA TB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed define the first address for the DMA access the TIMERx_DMA TB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMA TB, you will access the address of start address + 0x4. 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... In a word: Start Address = TIMERx_CTL0 + DMASAR*4

DMA transfer buffer register (TIMERx_DMATB)

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	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer* 4) will be accessed. The transfer Timer is calculated by hardware, and ranges from 0 to DMATC.

Channel input remap register(TIMERx_IRMP)(x=4)

Address offset: 0x50

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								CI3_RMP[1:0]		Reserved					

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:6	CI3_RMP[1:0]	Channel 3 input remap 00: Channel 3 input is connected to GPIO(TIMER4_CH3) 01: Channel 3 input is connected to IRC32K 10: Channel 3 input is connected to LXTAL 11: Channel 3 input is connected to RTC wakeup
5:0	Reserved	Must be kept at reset value

Configuration register (TIMERx_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
15:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	Reserved	Must be kept at reset value.

17.3. General level4 timer(TIMERx, x=15,16)

17.3.1. Overview

The general level4 timer module (TIMER15, TIMER16) is a one-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 general level4 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level4 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.

17.3.2. Characteristics

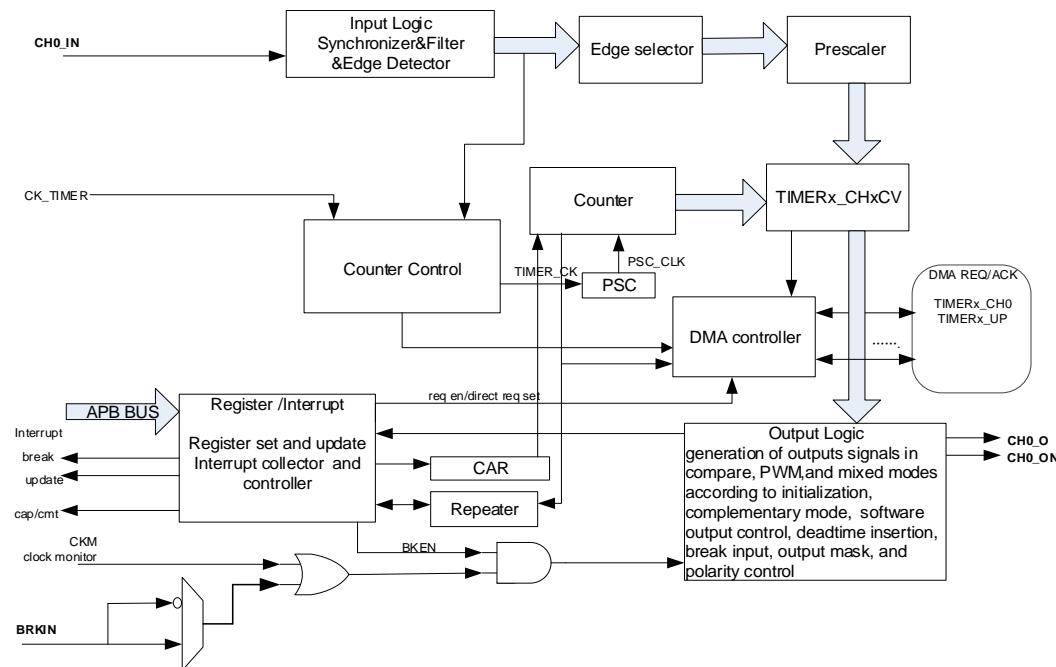
- Total channel num: 1.
- Counter width: 16 bits.
- Source of counter clock: internal clock.
- Counter modes: count up only.
- Programmable prescaler: 16 bit. The factor can be changed on the go.
- Each channel is user-configurable:
 - input capture mode, output compare mode, programmable PWM mode, single pulse mode
 - Programmable dead time insertion.
 - Auto reload function.
 - Programmable counter repetition function.
 - Break input.
 - Interrupt output or DMA request on: update, compare/capture event, and break input.

17.3.3. Block diagram

[Figure 17-53. General level4 timer block diagram](#) provides details of the internal

configuration of the general level4 timer.

Figure 17-53. General level4 timer block diagram



17.3.4. Function overview

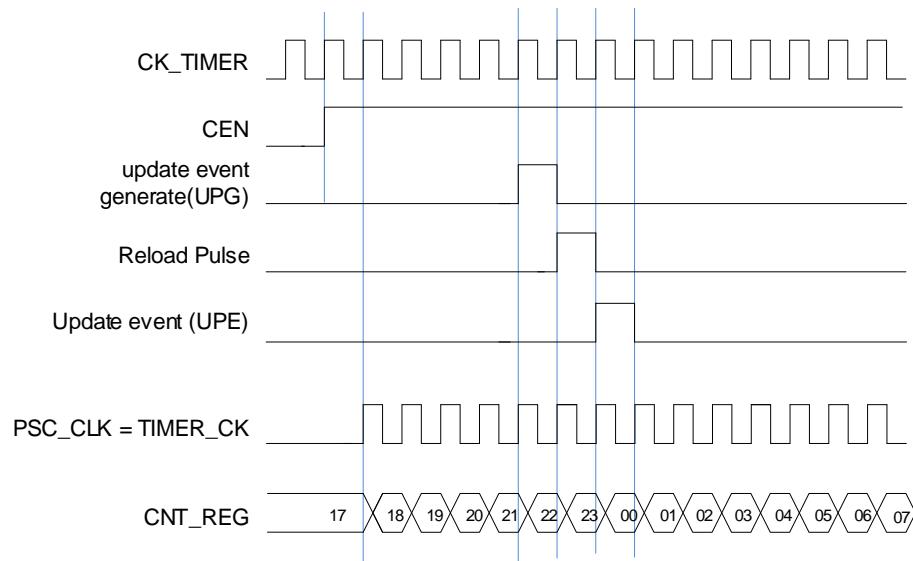
Clock selection

The general level4 TIMER can only being clocked by the CK_TIMER.

- Internal timer clock CK_TIMER which is from module RCU

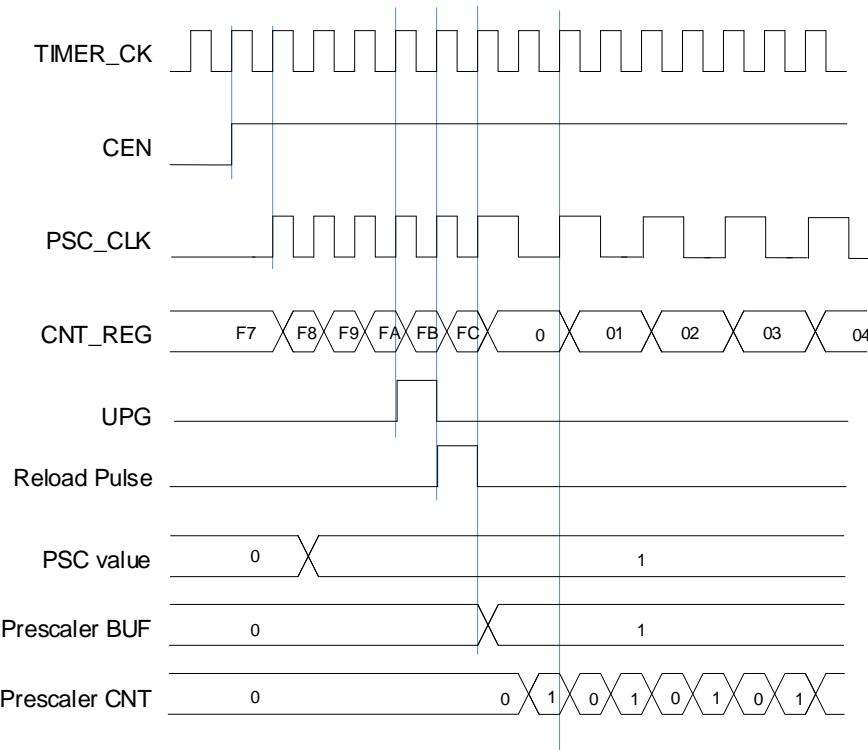
The general level4 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

Figure 17-54. Normal mode, internal clock divided by 1


Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed on the go but is taken into account at the next update event.

Figure 17-55. 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. If the repetition counter is set, the update events will be generated after (TIMERx_CREP+1) times of overflow. Otherwise the update event is generated each time when 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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

Figure 17-56. Up-counter timechart, PSC=0/1 show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x63.

Figure 17-56. Up-counter timechart, PSC=0/1

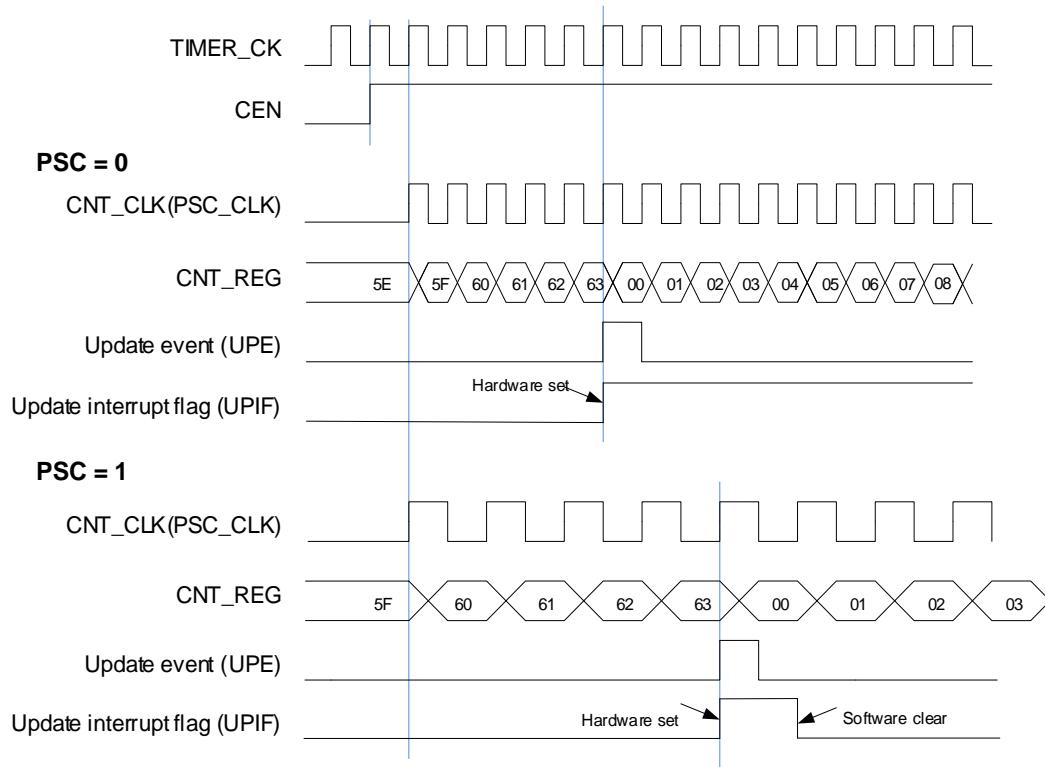
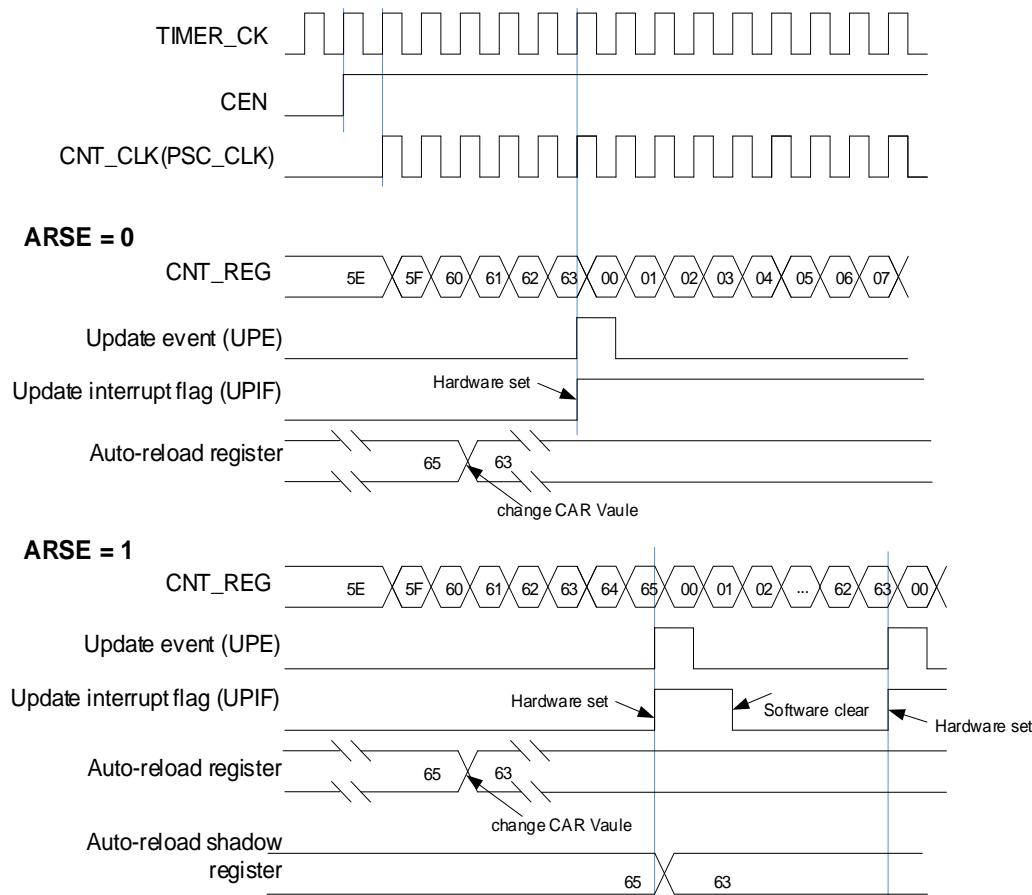


Figure 17-57. Up-counter timechart, change TIMERx_CAR on the go

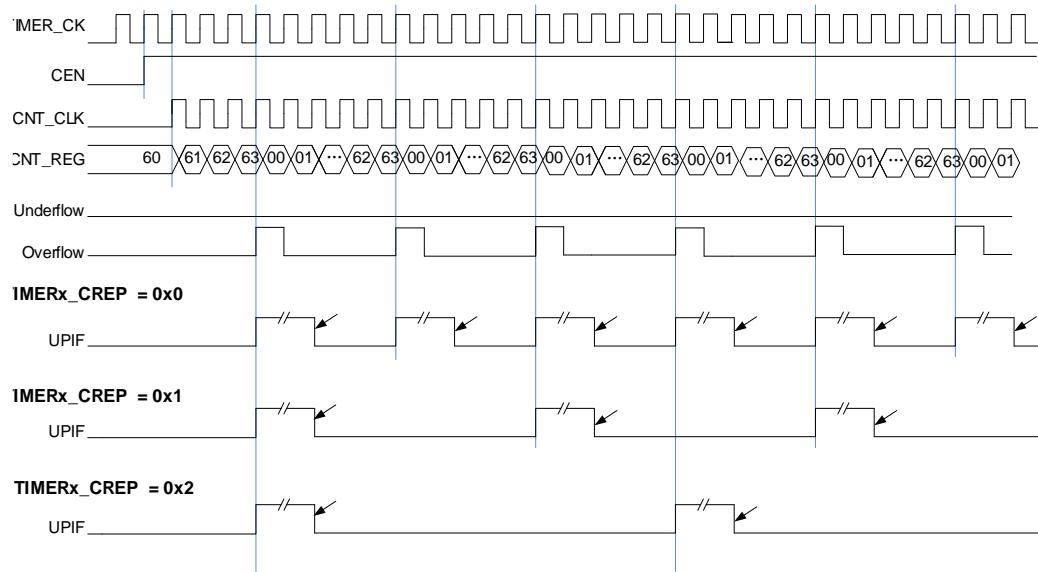


Repetition counter

Counter repetition is used to generate update events or update the timer registers only after a given number ($N+1$) of cycles of the counter, where N is CREP in TIMERx_CREP register. The repetition counter is decremented at each counter overflow in up-counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generate an update event.

Figure 17-58. Repetition timechart for up-counter

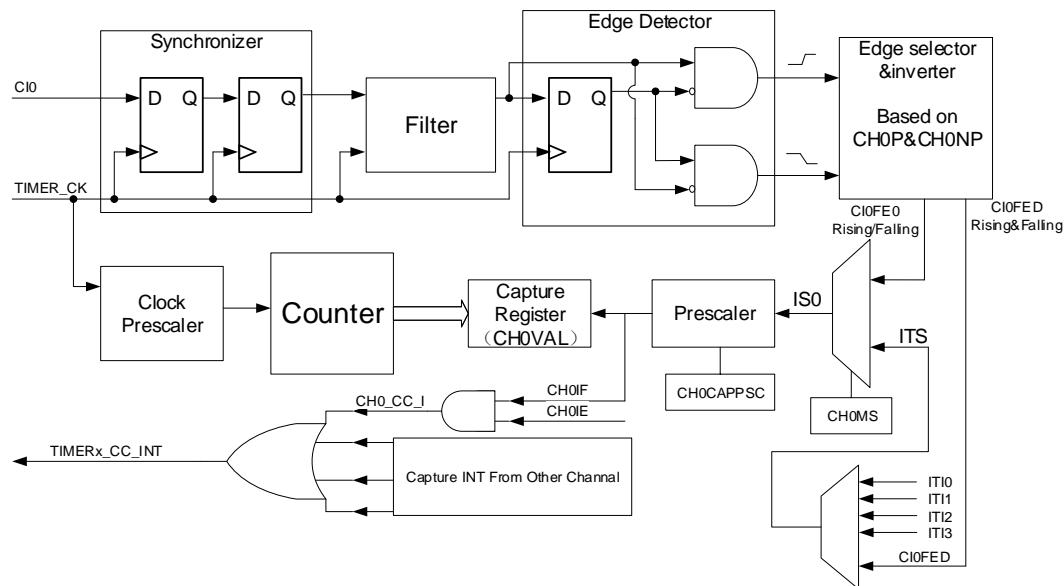


Capture/compare channels

The general level4 timer has one 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.

Input capture mode

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 enabled by **CHxIE** = 1.

Figure 17-59. Input capture logic


Channels' input signals (Clx) is the $TIMERx_CHx$ signal. First, the channel input signal (Clx) is synchronized to $TIMER_CK$ 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 by $CHxP$. One more selector is for the other channel and trig, controlled by $CHxMS$. The IC_prescaler make several the input event generate one effective capture event. On the capture event, $CHxVAL$ will restore the value of Counter.

So the process can be divided to several steps as below:

Step1: Filter configuration. ($CHxCAPFLT$ in $TIMERx_CHCTL0$)

Based on the input signal and requested signal quality, configure compatible $CHxCAPFLT$.

Step2: Edge selection. ($CHxP/CHxNP$ in $TIMERx_CHCTL2$)

Rising or falling edge, choose one by $CHxP/CHxNP$.

Step3: Capture source selection. ($CHxMS$ in $TIMERx_CHCTL0$)

As soon as you select one input capture source by $CHxMS$, you have set the channel to input mode ($CHxMS!=0x0$) and $TIMERx_CHxCV$ cannot be written any more.

Step4: Interrupt enable. ($CHxIE$ and $CHxDEN$ in $TIMERx_DMAINTEN$)

Enable the related interrupt enable; you can got the interrupt and DMA request.

Step5: Capture enables. ($CHxEN$ in $TIMERx_CHCTL2$)

Result: when you wanted input signal is got, $TIMERx_CHxCV$ will be set by counter's value. And $CHxIF$ is asserted. If the $CHxIF$ is high, the $CHxOF$ will be asserted also. The interrupt and DMA request will be asserted based on the configuration of $CHxIE$ and $CHxDEN$ in

TIMERx_DMAINTEN

Direct generation: if you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

Output compare mode

Figure 17-60. Output compare logic (with complementary output, x=0)

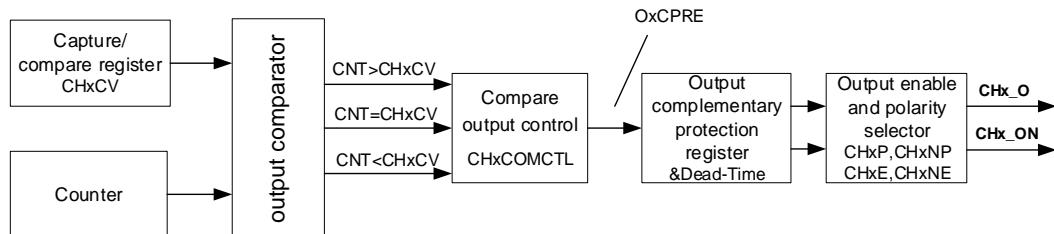


Figure 17-60. Output compare logic (with complementary output, x=0) show the logic circuit of output compare mode. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal (more details refer to [Channel output reference signal](#)) is described as below: The active level of OxCPRE is high, the output level of CHx_O/CHx_ON depends on OxCPRE signal, CHxP/CHxNP bit and CHxE/CHxNE bit (please refer to the TIMERx_CHCTL2 register for more details). For examples,

- 1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=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 CHxNP=0 (the active level of CHx_ON is low, contrary to OxCPRE), CHxNE=1 (the output of CHx_ON is enabled):
 - If the output of OxCPRE is active(high) level, the output of CHx_O is active(low) level;
 - If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(high) level.

When CH0_O and CH0_ON are output at the same time, the specific outputs of CH0_O and CH0_ON 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 CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be assert, if CHxDEN = 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 shadow enable mode by CHxCOMSEN
- * Set the output mode (Set/Clear/Toggle) by CHxCOMCTL.
- * Select the active high polarity by CHxP/CHxNP
- * Enable the output by CHxEN

Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN

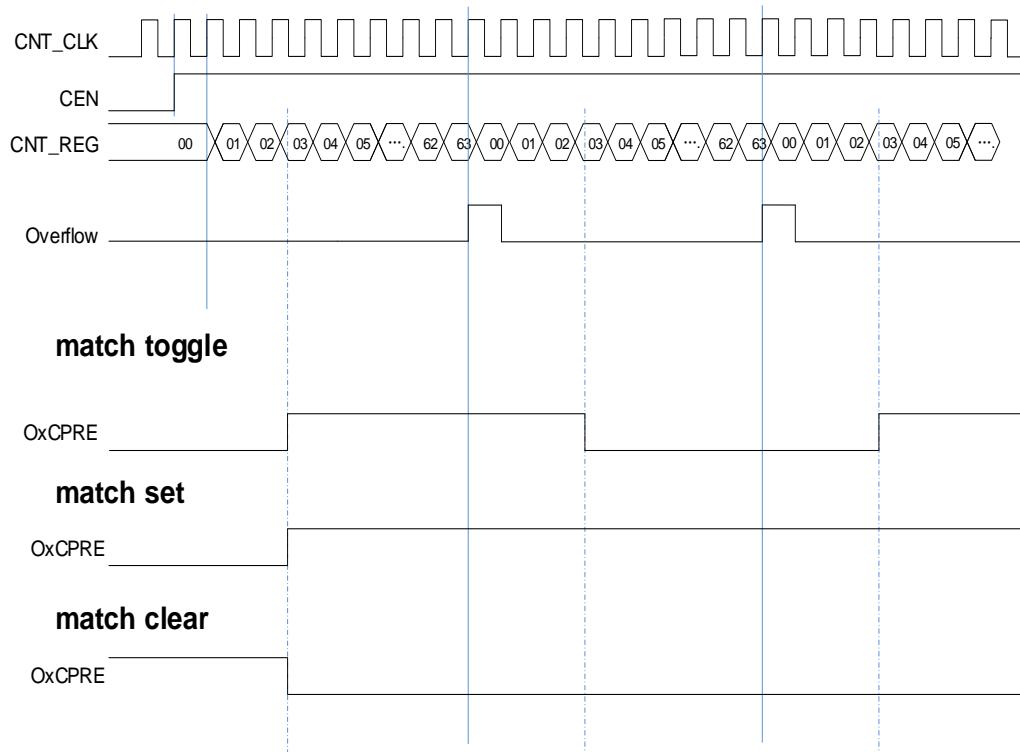
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV

About the CHxVAL; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

[**Figure 17-61. Output-compare under three modes**](#) show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 17-61. Output-compare under three modes



PWM mode

In the output PWM mode (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

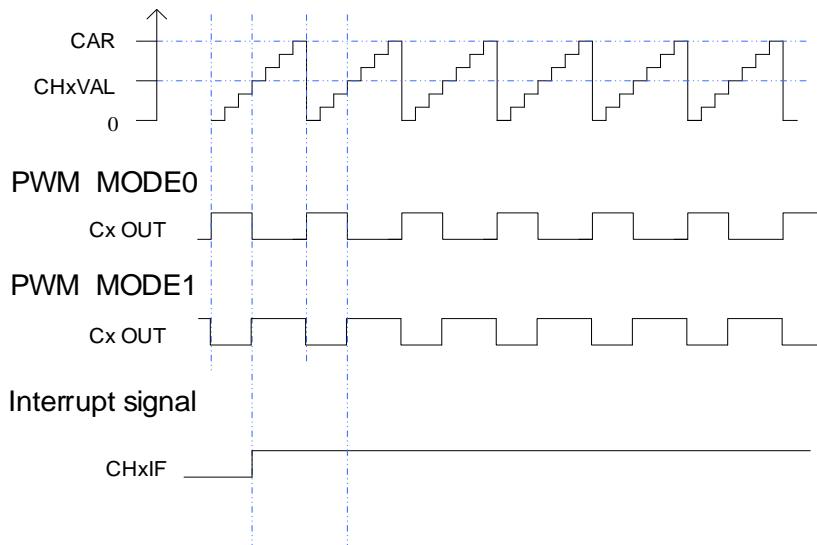
The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV.

[**Figure 17-62. PWM mode timechart**](#) shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).

And if `TIMERx_CHxCV` is equal to zero, the output will be always inactive under PWM mode0 (`CHxCOMCTL==3'b110`).

Figure 17-62. PWM mode timechart



Channel output reference signal

As is shown in [Figure 17-60. Output compare logic \(with complementary output, x=0\)](#). When the `TIMERx` is used in the compare match output mode, the `OxCPRE` signal (Channel x Output prepare signal) is defined by setting the `CHxCOMCTL` filed. The `OXCPRE` signal has several types of output function. These include, keeping the original level by setting the `CHxCOMCTL` field to `0x00`, set to 1 by setting the `CHxCOMCTL` field to `0x01`, set to 0 by setting the `CHxCOMCTL` field to `0x02` or signal toggle by setting the `CHxCOMCTL` field to `0x03` when the counter value matches the content of the `TIMERx_CHxCV` register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of `OxCPRE` output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the `OxCPRE` signal is a forced output which can be achieved by setting the `CHxCOMCTL` field to `0x04/0x05`. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the `TIMERx_CHxCV` values.

Outputs complementary

Function of complementary is for a pair of `CHx_O` and `CHx_ON`. Those two output signals cannot be active at the same time. The `TIMERx` has only 1 channel have this function. The complementary signals `CHx_O` and `CHx_ON` are controlled by a group of parameters: the

CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 17-7. Complementary outputs controlled by parameters

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output disable. If clock is enable: CHx_O = ISOx CHx_ON = ISOxN	
			1	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output disable.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON output enable. If clock is enable: CHx_O = ISOx CHx_ON = ISOxN	
		0	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON output disable.	
				1	CHx_O = LOW CHx_O output disable.	CHx_ON=OxCPRE⊕ CHxNP CHx_ON output enable
			1	0	CHx_O=OxCPRE⊕ CHxP CHx_O output enable	CHx_ON = LOW CHx_ON output disable.
				1	CHx_O=OxCPRE⊕ CHxP CHx_O output enable	CHx_ON=(!OxCPRE)⊕ CHxNP CHx_ON output enable
		1	0	0	CHx_O = CHxP CHx_O output disable.	CHx_ON = CHxNP CHx_ON output disable.
				1	CHx_O = CHxP CHx_O output enable	CHx_ON=OxCPRE⊕ CHxNP CHx_ON output enable
			1	0	CHx_O=OxCPRE⊕ CHxP CHx_O output enable	CHx_ON = CHxNP CHx_ON output enable.
				1	CHx_O=OxCPRE⊕ CHxP CHx_O output enable	CHx_ON=(!OxCPRE)⊕ CHxNP CHx_ON output enable.

Dead time insertion

The dead time insertion is enabled when both CHxEN and CHxNEN are 1'b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for channel 1. The detail about the delay time, refer to the register TIMERx_CCHP.

The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

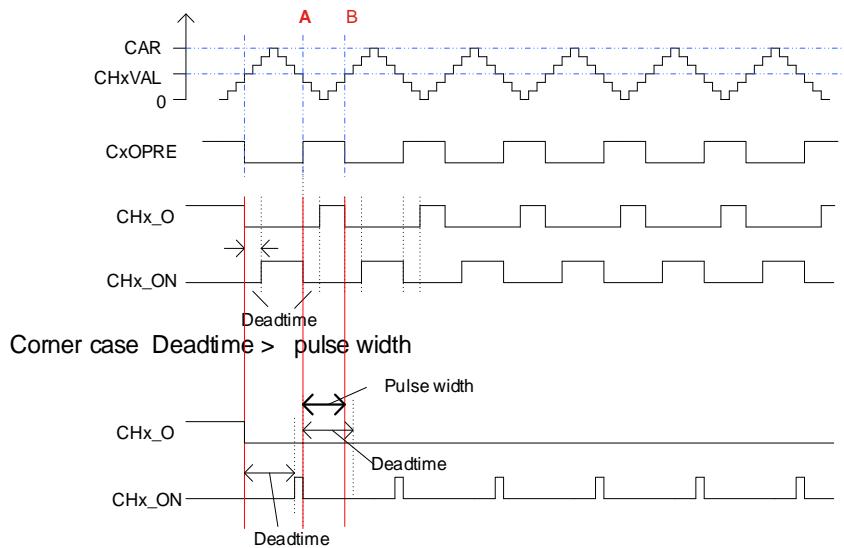
When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the [Figure 17-63. Complementary output with dead-time insertion](#), CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, at point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value 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 CHx_O duty cycle, then the CHx_O signal is always the inactive value. (as show in the [Figure 17-63. Complementary output with dead-time insertion](#).)

The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 17-63. Complementary output with dead-time insertion.



Break function

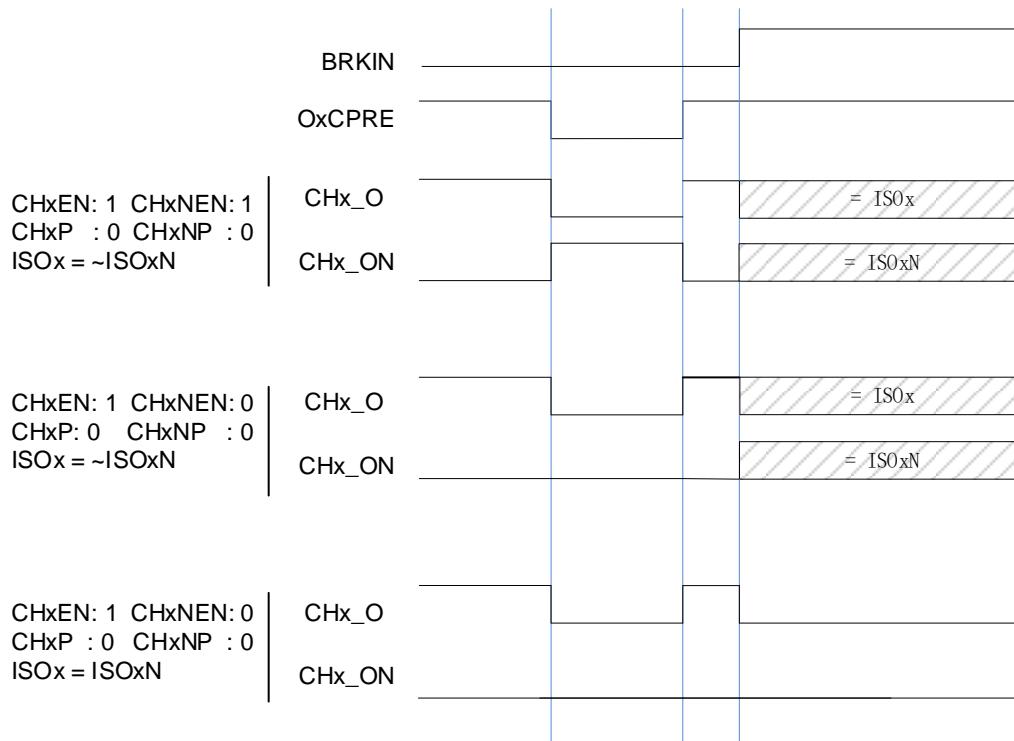
In this function, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs. The break sources are input break pin

and HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead-time.

When a break occurs, the BRKIF bit in the TIMERx_INTF register is set. If BRKIE is 1, an interrupt generated.

Figure 17-64. Output behavior in response to a break(The break high active)



Single pulse mode

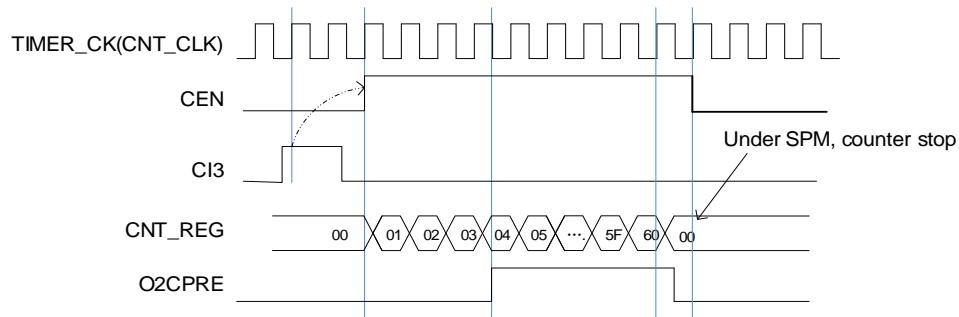
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event automatically. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the CEN bit to 1 using software. Setting the CEN bit to 1 or a trigger from the trigger signals 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 using software, the counter will be stopped and its value 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 trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0/1 register. After a trigger rising occurs in the single pulse mode, the OxCOPRE signal will immediately be forced to the state which the OxCOPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 17-65. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Of course, you have to enable a DMA request which will be asserted by some internal event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB 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), then the timer's DMA request is finished. 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, DMATA+0xc at the next 3 accesses to TIMERx_DMATB. In one word, one time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex™-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

17.3.5. **TIMERx registers(x=15, 16)**

TIMER15 secure access base address: 0x5001 8000

TIMER15 non-secure access base address: 0x4001 8000

TIMER16 secure access base address: 0x5001 8400

TIMER16 non-secure access base address: 0x4001 8400

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Reserved				CKDIV[1:0]		ARSE		Reserved		SPM	UPS	UPDIS	CEN

rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:10	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 the timer clock (TIMER_CK) and the dead-time and sampling clock (DTS), which is used by the dead-time generators and the digital filters. 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 TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value
3	SPM	Single pulse mode. 0: Counter continues after update event. 1: The CEN 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 generate an update interrupt or 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 DMA request.

1	UPDIS	<p>Update disable.</p> <p>This bit is used to enable or disable the update event generation.</p> <p>0: update event enable. The update event is generate and the buffered registers are loaded with their preloaded values when one of the following events occurs:</p> <p>The UPG bit is set</p> <p>The counter generates an overflow or underflow event</p> <p>The slave mode controller generates an update event.</p> <p>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 if the slave mode controller generates a hardware reset event.</p>
0	CEN	<p>Counter enable</p> <p>0: Counter disable</p> <p>1: Counter enable</p> <p>The CEN bit must be set by software when timer works in external clock, pause mode and encoder mode. While in event mode, the hardware can set the CEN bit automatically.</p>

Control register 1 (TIMERx_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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					ISOON	ISO0	Reserved			DMAS	CCUC	Reserved	CCSE		
					rw	rw				rw	rw				

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9	ISOON	<p>Idle state of channel 0 complementary output</p> <p>0: When POEN bit is reset, CH0_ON is set low.</p> <p>1: When POEN bit is reset, CH0_ON is set high</p> <p>This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.</p>
8	ISO0	<p>Idle state of channel 0 output</p> <p>0: When POEN bit is reset, CH0_O is set low.</p>

1: When POEN bit is reset, CH0_O is set high

The CH0_O output changes after a dead-time if CH0_ON is implemented. This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.

7:4	Reserved	Must be kept at reset value
3	DMAS	DMA request source selection 0: DMA request of channel x is sent when capture/compare event occurs. 1: DMA request of channel x is sent when update event occurs.
2	CCUC	Commutation control shadow register update control When the commutation control shadow enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by when CMTG bit is set or a rising edge of TRGI occurs. 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, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. When a channel does not have a complementary output, this bit has no effect.

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CH0DEN	UPDEN	BRKIE	Reserved	CMTIE	Reserved			CHOIE	UPIE	
rw					rw	rw	rw		rw	rw			rw	rw	

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9	CH0DEN	Channel 0 capture/compare DMA request enable

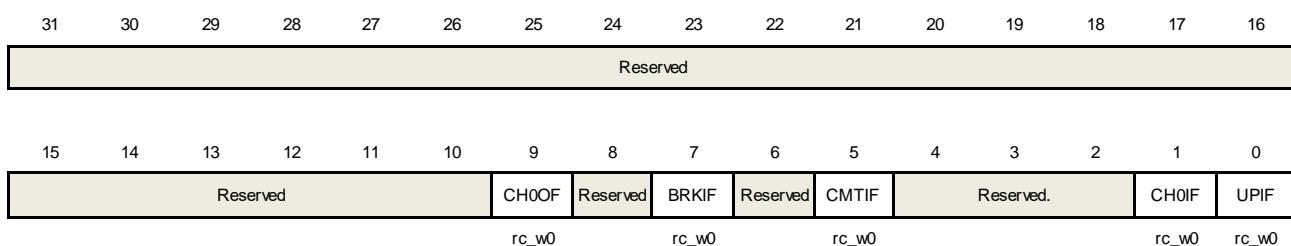
		0: disabled 1: enabled
8	UPDEN	Update DMA request enable 0: disabled 1: enabled
7	BRKIE	Break interrupt enable 0: disabled 1: enabled
6	Reserved	Must be kept at reset value
5	CMTIE	Commutation interrupt enable 0: disabled 1: enabled
4:2	Reserved	Must be kept at reset value
1	CH0IE	Channel 0 capture/compare 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
31:10	Reserved	Must be kept at reset value
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	<p>Break interrupt flag</p> <p>This flag is set by hardware when the break input goes active, and cleared by software if the break input is not active.</p> <p>0: No active level break has been detected. 1: An active level has been detected.</p>
6	Reserved	Must be kept at reset value
5	CMTIF	<p>Channel commutation interrupt flag</p> <p>This flag is set by hardware when channel's commutation event occurs, and cleared by software</p> <p>0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred</p>
4:2	Reserved	Must be kept at reset value
1	CH0IF	<p>Channel 0's capture/compare interrupt flag</p> <p>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.</p> <p>0: No Channel 0 interrupt occurred 1: Channel 0 interrupt occurred</p>
0	UPIF	<p>Update interrupt flag</p> <p>This bit is set by hardware on an update event and cleared by software.</p> <p>0: No update interrupt occurred 1: Update interrupt occurred</p>

Software event generation register (TIMERx_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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRKG	Reserved	CMTG	Reserved			CH0G	UPG

Bits	Fields	Descriptions
------	--------	--------------

31:8	Reserved	Must be kept at reset value
7	BRKG	<p>Break event generation</p> <p>This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is set, related interrupt or DMA transfer can occur if enabled.</p> <p>0: No generate a break event 1: Generate a break event</p>
6	Reserved	Must be kept at reset value
5	CMTG	<p>Channel commutation event generation</p> <p>This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1).</p> <p>0: No affect 1: Generate channel's c/c control update event</p>
4:2	Reserved	Must be kept at reset value
1	CH0G	<p>Channel 0's capture or compare event generation</p> <p>This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high.</p> <p>0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event</p>
0	UPG	<p>Update event generation</p> <p>This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time.</p> <p>0: No generate an update event 1: Generate an update event</p>

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								Reserved	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]		CH0CAPFLT[3:0]		CH0CAPPSC[1:0]

rw

rw

rw

Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6:4	CH0COMCTL[2:0]	<p>Channel 0 compare output control</p> <p>This bit-field controls the behavior of the output reference signal O0CPRE which drives CH0_O and CH0_ON. O0CPRE is active high, while CH0_O and CH0_ON active level depends on CH0P and CH0NP bits.</p> <p>000: Frozen. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT.</p> <p>001: Set the channel output. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV.</p> <p>010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV.</p> <p>011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV.</p> <p>100: Force low. O0CPRE is forced low level.</p> <p>101: Force high. O0CPRE is forced high level.</p> <p>110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active.</p> <p>111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive.</p> <p>When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from "frozen" mode to "PWM" mode or when the result of the comparison changes.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 00(COMPARE MODE).</p>
3	CH0COMSEN	<p>Channel 0 compare output shadow enable</p> <p>When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled.</p> <p>0: Channel 0 output compare shadow disable</p> <p>1: Channel 0 output compare shadow enable</p> <p>The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is</p>

		11 and CH0MS bit-field is 00.
2	CH0COMFEN	<p>Channel 0 output compare fast enable</p> <p>When this bit is set, the effect of an event on the trigger input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison.</p> <p>0: Channel 0 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles.</p> <p>1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.</p>
1:0	CH0MS[1:0]	<p>Channel 0 I/O mode selection</p> <p>This bit-field specifies the work mode of the channel and the input signal selection.</p> <p>This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).).</p> <p>00: Channel 0 is configured as output</p> <p>01: Channel 0 is configured as input, IS0 is connected to CI0FE0</p> <p>10: Reserved</p> <p>11: Channel 0 is configured as input, IS0 is connected to ITS. This mode is working only if an internal trigger input is selected through TRGS bits in TIMERx_SMCFG register.</p>

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:4	CH0CAPFLT[3:0]	<p>Channel 0 input capture filter control</p> <p>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.</p> <p>0000: Filter disabled, $f_{SAMP}=f_{DTS}$, $N=1$</p> <p>0001: $f_{SAMP}=f_{CK_TIMER}$, $N=2$</p> <p>0010: $f_{SAMP}=f_{CK_TIMER}$, $N=4$</p> <p>0011: $f_{SAMP}=f_{CK_TIMER}$, $N=8$</p> <p>0100: $f_{SAMP}=f_{DTS}/2$, $N=6$</p> <p>0101: $f_{SAMP}=f_{DTS}/2$, $N=8$</p> <p>0110: $f_{SAMP}=f_{DTS}/4$, $N=6$</p> <p>0111: $f_{SAMP}=f_{DTS}/4$, $N=8$</p> <p>1000: $f_{SAMP}=f_{DTS}/8$, $N=6$</p> <p>1001: $f_{SAMP}=f_{DTS}/8$, $N=8$</p> <p>1010: $f_{SAMP}=f_{DTS}/16$, $N=5$</p> <p>1011: $f_{SAMP}=f_{DTS}/16$, $N=6$</p> <p>1100: $f_{SAMP}=f_{DTS}/16$, $N=8$</p>

		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 clear. 00: Prescaler disable, capture is done on each channel input edge 01: Capture is done every 2 channel input edges 10: Capture is done every 4channel input edges 11: Capture is done every 8 channel input edges
1:0	CH0MS[1:0]	Channel 0 mode selection Same as Output compare mode

Channel control register 2 (TIMERx_CHCTL2)

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

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, In conjunction with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled

1: Channel 0 complementary output enabled

1	CH0P	<p>Channel 0 capture/compare function polarity</p> <p>When channel 0 is configured in output mode, this bit specifies the output signal polarity.</p> <p>0: Channel 0 active high</p> <p>1: Channel 0 active low</p> <p>When channel 0 is configured in input mode, this bit specifies the Cl0 signal polarity.</p> <p>[CH0NP, CH0P] will select the active trigger or capture polarity for Cl0FE0 or Cl1FE0.</p> <p>[CH0NP==0, CH0P==0]: ClxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will not be inverted.</p> <p>[CH0NP==0, CH0P==1]: ClxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And ClxFE0 will be inverted.</p> <p>[CH0NP==1, CH0P==0]: Reserved.</p> <p>[CH0NP==1, CH0P==1]: ClxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And ClxFE0 will be not inverted.</p> <p>This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.</p>
0	CH0EN	<p>Channel 0 capture/compare function enable</p> <p>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.</p> <p>0: Channel 0 disabled</p> <p>1: Channel 0 enabled</p>

Counter register (TIMERx_CNT)

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															
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 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)

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

rw

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 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
CARL[15:0]															

rw

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)

31	30	29	28	27	26	25	24	23	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								CREP[7:0]							

rw

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 counting down to zero, an update event is 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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-field indicates the counter value corresponding to 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 every update event.

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	POEN	<p>Primary output enable</p> <p>This bit is set by software or automatically by hardware depending on the OAEN bit.</p> <p>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 CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN in TIMERx_CHCTL2 register) have been set.</p> <p>0: Channel outputs are disabled or forced to idle state.</p> <p>1: Channel outputs are enabled.</p>
14	OAEN	<p>Output automatic enable</p> <p>This bit specifies whether the POEN bit can be set automatically by hardware.</p> <p>0: POEN can be set by hardware.</p> <p>1: POEN can be set by hardware automatically at the next update event, if the break input is not active.</p> <p>This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.</p>
13	BRKP	<p>Break polarity</p> <p>This bit specifies the polarity of the BRKIN input signal.</p> <p>0: BRKIN input active low</p> <p>1: BRKIN input active high</p>
12	BRKEN	<p>Break enable</p> <p>This bit can be set to enable the BRKIN and CCS clock failure event inputs.</p> <p>0: Break inputs disabled</p> <p>1: Break inputs enabled</p> <p>This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.</p>
11	ROS	Run mode off-state configure

When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode.

0: When POEN bit is set, the channel output signals (CHx_O/CHx_ON) are disabled.

1: When POEN bit is set, the channel output signals (CHx_O/CHx_ON) are enabled, with relationship to CHxEN/CHxNEN bits in TIMERx_CHCTL2 register.

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 configure
		<p>When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode.</p> <p>0: When POEN bit is reset, the channel output signals (CHx_O/CHx_ON) are disabled.</p> <p>1: When POEN bit is reset, the channel output signals (CHx_O/CHx_ON) are enabled, with relationship to CHxEN/CHxNEN bits in TIMERx_CHCTL2 register.</p> <p>This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.</p>
9:8	PROT[1:0]	Complementary register protect control
		<p>This bit-field specifies the write protection property of registers.</p> <p>00: protect disable. No write protection.</p> <p>01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected.</p> <p>10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected.</p> <p>11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/CHxCOMSEN bits in TIMERx_CHCTL0/1 registers (if the related channel is configured in output) are writing protected.</p> <p>This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.</p>
7:0	DTCFG[7:0]	Dead time configure
		<p>This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow :</p> <p>DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}.</p> <p>DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0])x t_{DT}, t_{DT}=t_{DTS}*2.</p> <p>DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0])x t_{DT}, t_{DT}=t_{DTS}*8.</p> <p>DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0])x t_{DT}, t_{DT}=t_{DTS}*16.</p> <p>This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.</p>

DMA configuration register (TIMERx_DMACFG)

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

rw

rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This field is defined the number of DMA will access(R/W) the register of TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This field define the first address for the DMA access the TIMERx_DMATB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMATB, you will access the address of start address + 0x4. 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... In a word: Start Address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

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

rw

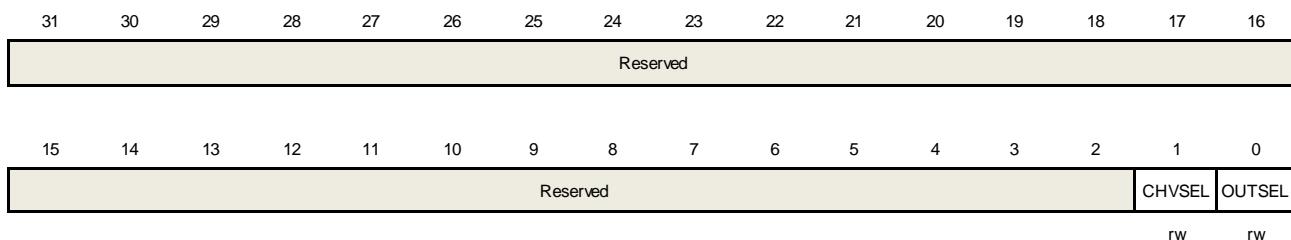
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer * 4) will be accessed. The transfer Timer is calculated by hardware, 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)



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	OUTSEL	The output value selection This bit-field set and reset by software 1: If POEN and IOS is 0, the output disabled 0: No effect

17.4. Basic timer (TIMERx, x=5)

17.4.1. Overview

The basic timer module (Timer5) reference is a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate DMA request.

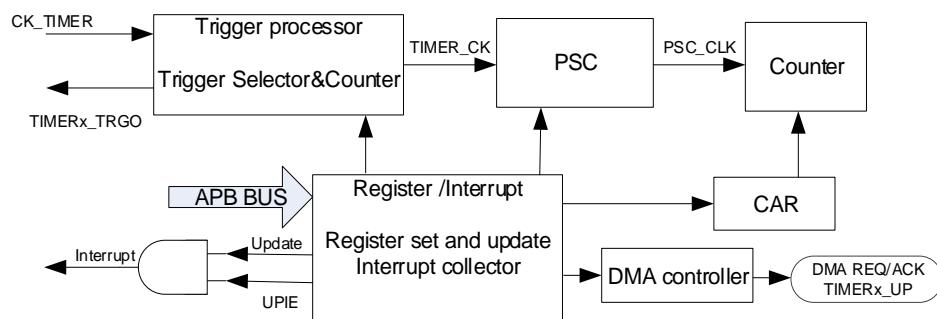
17.4.2. Characteristics

- Counter width: 16bit.
 - Source of count clock is internal clock only.
 - Multiple counter modes: count up.
 - Programmable prescaler: 16 bit. Factor can be changed on the go.
 - Auto-reload function.
 - Interrupt output or DMA request on update event.

17.4.3. Block diagram

Figure 17-66. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 17-66. Basic timer block diagram

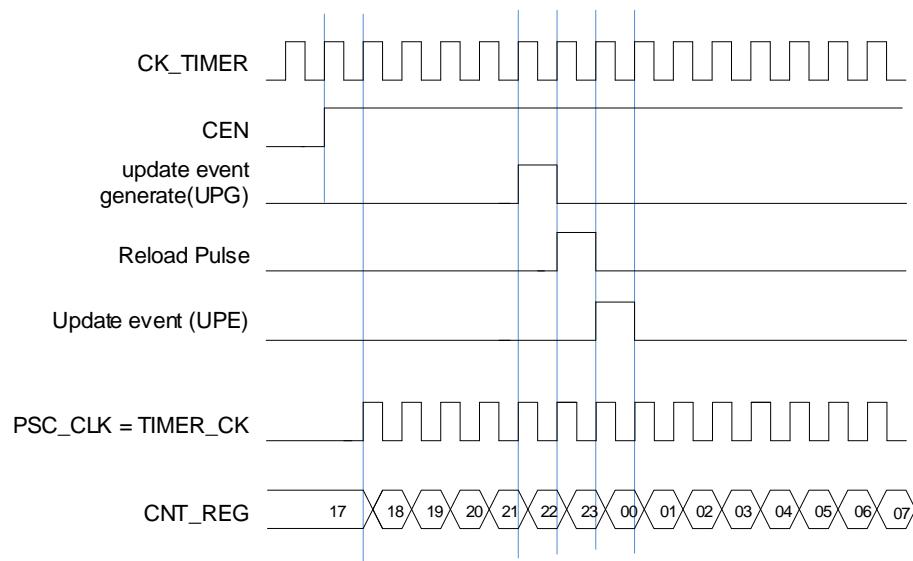


17.4.4. Function overview

Clock selection

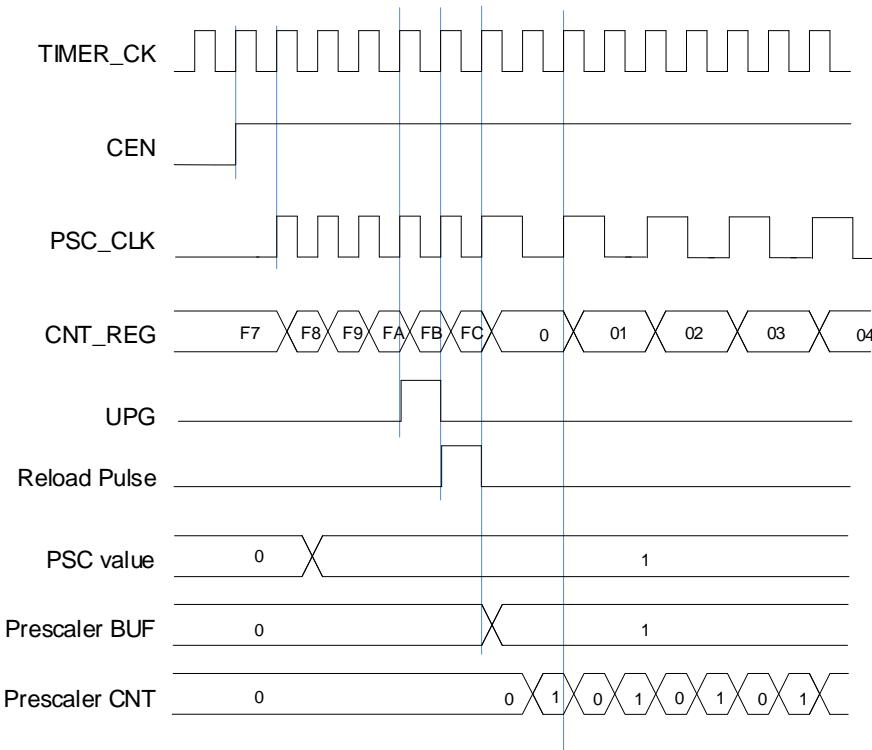
The basic TIMER can only be clocked by the internal timer clock CK_TIMER, which is from the source named CK_TIMER in RCU

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

Figure 17-67. Normal mode, internal clock divided by 1


Prescaler

The prescaler can divide the timer clock (TIMER_CK) to the counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled through prescaler register (TIMERx_PSC) which can be changed on the go but be taken into account at the next update event.

Figure 17-68. 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 to count once again from 0. The update event is generated at each counter overflow. The counting direction bit DIR in the TIMERx_CTL1 register should be set to 0 for the up counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the registers (repetition counter, auto reload register, prescaler register) are updated.

The following figures show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x63.

Figure 17-69. Timing chart of up counting mode, PSC=0/1

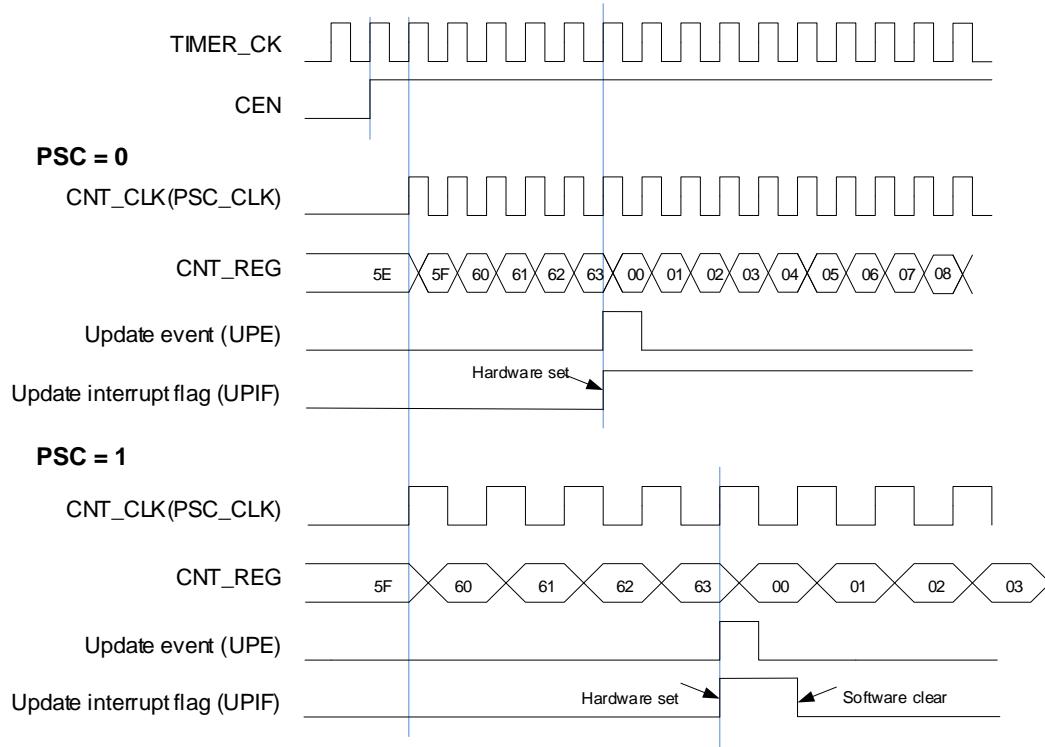
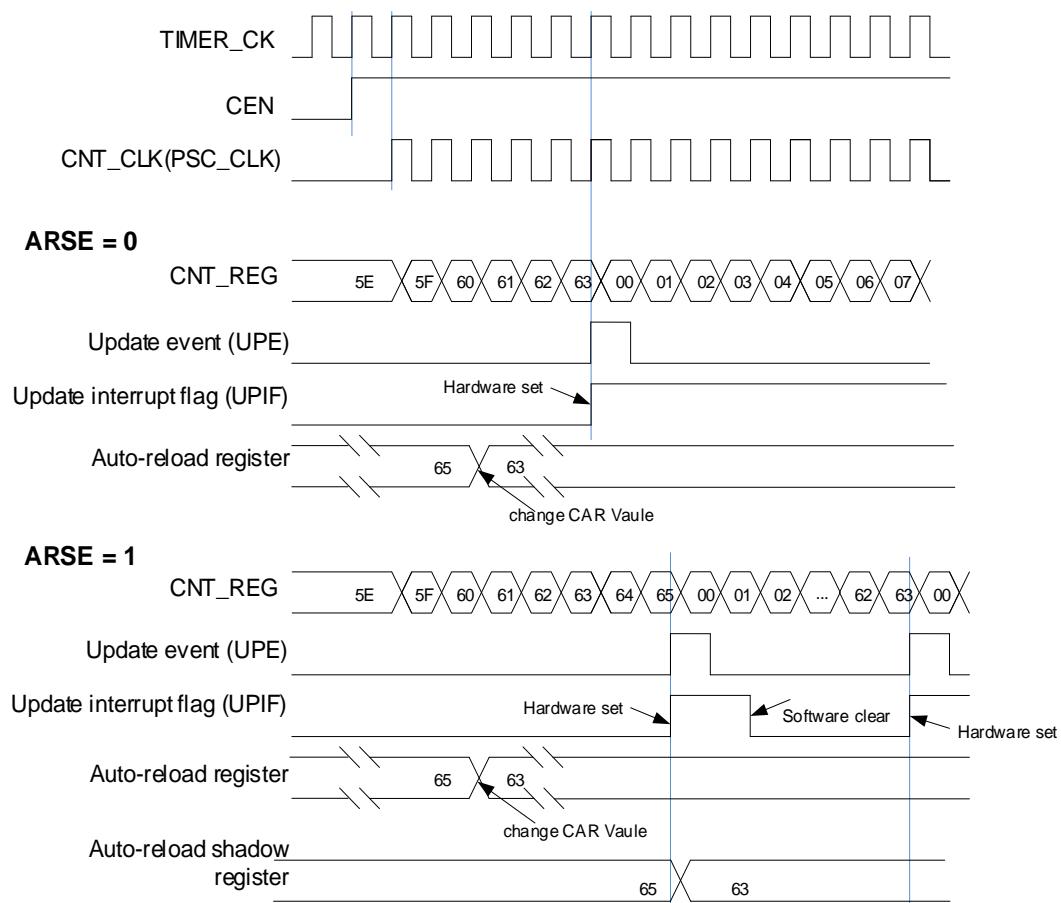


Figure 17-70. Timing chart of up counting mode, change TIMERx_CAR ongoing



Timer debug mode

When the Cortex™-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

17.4.5. **TIMERx registers(x=5)**

TIMER5 secure access base address: 0x5000 1000

TIMER5 non-secure access base address: 0x4000 1000

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ARSE	Reserved		SPM	UPS	UPDIS	CEN	
								rw			rw	rw	rw	rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Counter continues after update event. 1: The CEN 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: When enabled, any of the following events generate an update interrupt or DMA request: The UPG bit is set The counter generates an overflow or underflow event The slave mode controller generates an update event. 1: When enabled, only counter overflow/underflow generates an update interrupt or 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 if the slave mode controller generates a hardware reset event.

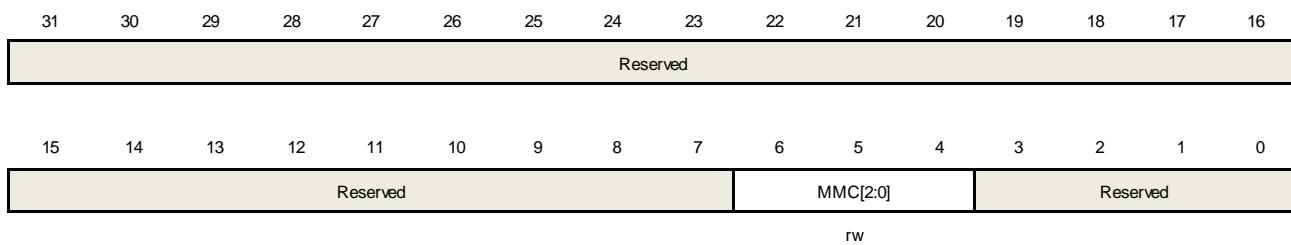
0	CEN	<p>Counter enable</p> <p>0: Counter disable</p> <p>1: Counter enable</p> <p>The CEN bit must be set by software when timer works in external clock, pause mode and encoder mode. While in event mode, the hardware can set the CEN bit automatically.</p>
---	-----	---

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
31:7	Reserved	Must be kept at reset value.
6:4	MMC[2:0]	<p>Master mode control</p> <p>These bits control the selection of TRGO signal, which is sent in master mode to slave timers for synchronization function.</p> <p>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.</p> <p>001: Enable. This mode is useful to start several timers at the same time or to control a window in which a slave timer is enabled. In this mode the master mode controller selects the counter enable signal TIMERx_EN 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.</p> <p>010: Update. In this mode the master mode controller selects the update event as TRGO.</p>

3:0 Reserved Must be kept at reset value.

Interrupt enable register (TIMERx_DMANTEN)

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	
Reserved								UPDEN	Reserved							

rw

rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: disabled 1: enabled
7:1	Reserved	Must be kept at reset value.
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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												UPIF	rc_w0		

rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag

This bit is set by hardware on an update event and cleared by software.

0: No update interrupt occurred

1: Update interrupt occurred

Software event generation register (TIMERx_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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	<p>This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time.</p> <p>0: No generate an update event</p> <p>1: Generate an update event</p>

Counter register (TIMERx_CNT)

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															
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 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).

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

rw

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 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
CARL[15:0]															

rw

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.

18. Universal synchronous/asynchronous receiver/transmitter (USART)

18.1. Overview

The Universal Synchronous/Asynchronous Receiver/Transmitter (USART) provides a flexible serial data exchange interface. Data frames can be transferred in full duplex or half duplex mode, synchronously or asynchronously through this interface. A programmable baud rate generator divides the UCLK (PCLK, CK_SYS, LXTAL, IRC16M) to produce a dedicated wide range baudrate clock for the USART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the USART implements several other types of serial data exchange modes, such as IrDA (infrared data association) SIR mode, smartcard mode, LIN (local interconnection network) mode, half-duplex mode and synchronous mode. It also supports multiprocessor communication mode, and hardware flow control protocol (CTS/RTS). The data frame can be transferred from LSB or MSB bit. The polarity of the TX/RX pins can be configured independently and flexibly.

All USARTs support DMA function for high-speed data communication.

18.2. Characteristics

- NRZ standard format
- Asynchronous, full duplex communication
- Half duplex single wire communications
- Receive FIFO function
- Dual clock domain:
 - Asynchronous PCLK and USART clock independent of PCLK
 - Baud rate programming independent from the UCLK reprogramming
- Programmable baud-rate generator allowing speed up to 11.25 MBit/s when the clock frequency is 90 MHz and oversampling is by 8
- Fully programmable serial interface characteristics:
 - A data word (8 or 9 bits) LSB or MSB first
 - Even, odd or no-parity bit generation/detection
 - 0.5, 1, 1.5 or 2 stop bit generation
- Swappable Tx/Rx pin
- Configurable data polarity
- Auto baud rate detection
- Hardware Modem operations (CTS/RTS) and RS485 drive enable
- Configurable multibuffer communication using centralized DMA
- Separate enable bits for Transmitter and Receiver

- Parity control
 - Transmits parity bit
 - Checks parity of received data byte
- LIN Break generation and detection
- IrDA Support
- Synchronous mode and transmitter clock output for synchronous transmission
- ISO 7816-3 compliant smartcard interface
 - Character mode ($T=0$)
 - Block mode ($T=1$)
 - Direct and inverse convention
- Multiprocessor communication
 - Enter into mute mode if address match does not occur
 - Wake up from mute mode by idle line or address mark detection
- Support for ModBus communication
 - Timeout feature
 - CR/LF character recognition
- Wake up from Deep-sleep mode
 - By standard RBNE interrupt
 - By WUF interrupt
- Various status flags
 - Flags for transfer detection: Receive buffer not empty (RBNE), receive FIFO full (RFF), Transmit buffer empty (TBE), transfer complete (TC).
 - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR)
 - Flag for hardware flow control: CTS changes (CTSF)
 - Flag for LIN mode: LIN break detected (LBDF)
 - Flag for multiprocessor communication: IDLE frame detected (IDLEF)
 - Flag for ModBus communication: Address/character match (AMF) and receiver timeout (RTF)
 - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF)
 - Wakeup from Deep-sleep mode flag
 - Interrupt occurs at these events when the corresponding interrupt enable bits are set

While USART0 and USART2 is fully implemented, USART1 is only partially implemented with the following features not supported.

- Auto baud rate detection
- Smartcard mode
- IrDA SIR ENDEC block
- LIN mode
- Dual clock domain and wakeup from Deep-sleep mode
- Receiver timeout interrupt
- ModBus communication

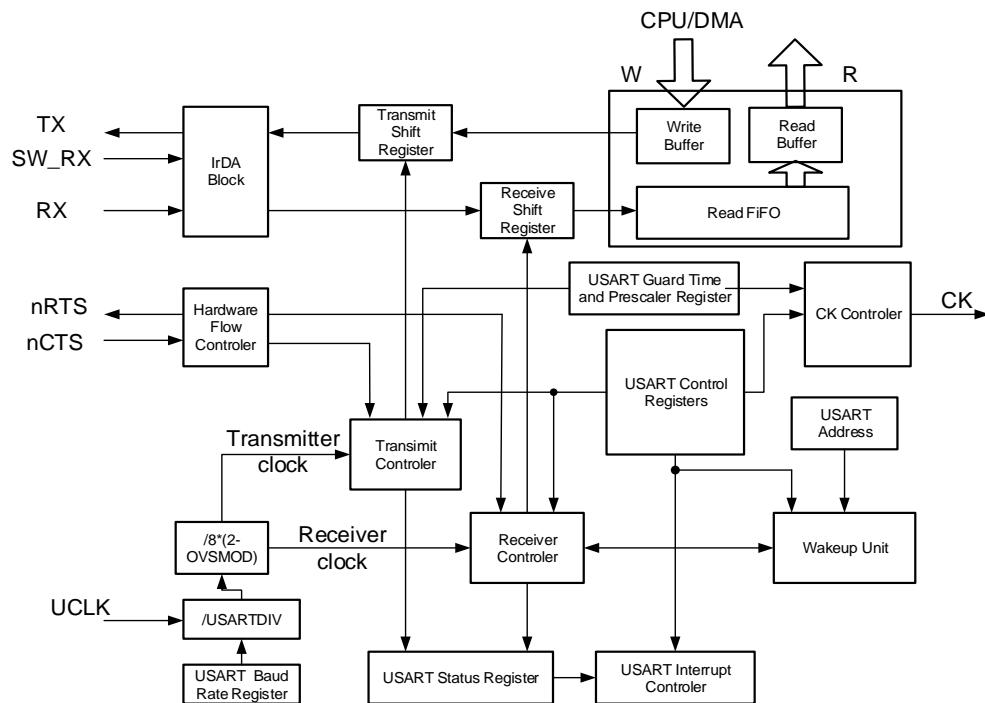
18.3. Function overview

The interface is externally connected to another device by the main pins listed in [Table 18-1. Description of USART important pins](#).

Table 18-1. Description of USART important pins

Pin	Type	Description
RX	Input	Receive Data
TX	Output I/O (single-wire/smartcard mode)	Transmit Data. High level When enabled but nothing to be transmitted
CK	Output	Serial clock for synchronous communication
nCTS	Input	Clear to send in Hardware flow control mode
nRTS	Output	Request to send in Hardware flow control mode

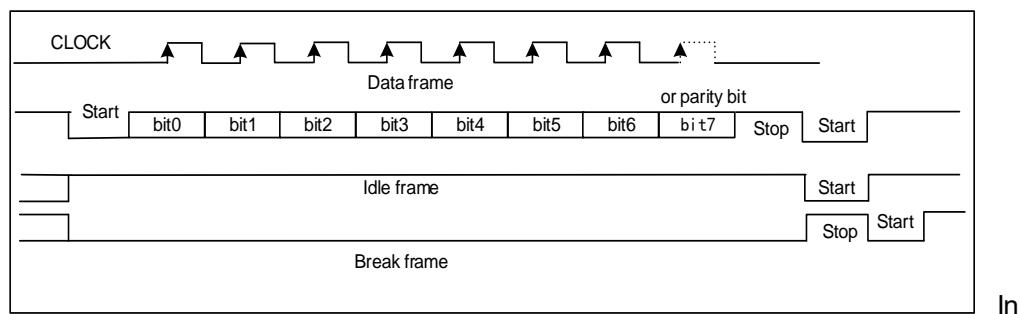
Figure 18-1. USART module block diagram



18.3.1. USART frame format

The USART frame starts with a start bit and ends up with a number of stop bits. The length of the data frame is configured by the WL bit in the USART_CTL0 register. The last data bit can be used as parity check bit by setting the PCEN bit of in USART_CTL0 register. When the WL bit is reset, the parity bit is the 7th bit. When the WL bit is set, the parity bit is the 8th bit. The method of calculating the parity bit is selected by the PM bit in USART_CTL0 register.

Figure 18-2. USART 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 USART_CTL1 register.

Table 18-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

18.3.2. Baud rate generation

The baud-rate divider is a 16-bit number which consists of a 12-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 USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the USART clock:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (17-1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (17-2)$$

The USART clock must be enabled through the clock control unit before enabling the USART.

For example, when oversampled by 16:

1. Get USARTDIV by calculating the value of USART_BAUD:

If USART_BAUD=0x21D, then INTDIV=33 (0x21), FRADIV=13 (0xD).

USARTDIV=33+13/16=33.81.

2. Get the value of USART_BAUD by calculating the value of USARTDIV:

If USARTDIV=30.37, then INTDIV=30 (0x1E).

$16 \times 0.37 = 5.92$, the nearest integer is 6, so FRADIV=6 (0x6).

USART_BAUD=0x1E6.

Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

18.3.3. USART transmitter

If the transmit enable bit (TEN) in USART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shifts out the transmit data frame through the TX pin. The polarity of the TX pin can be configured by the TINV bit in the USART_CTL1 register. Clock pulses can 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.

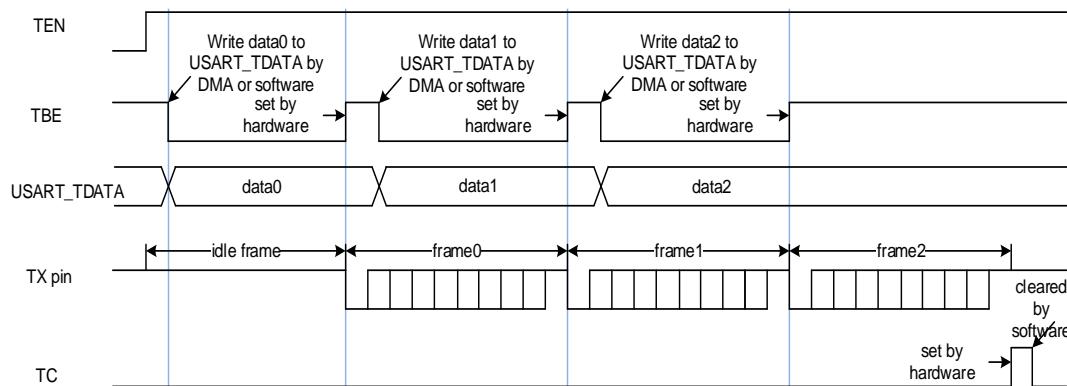
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_TDATA 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 USART_TDATA 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 USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in [Figure 18-3. USART transmit procedure](#). The software operating process is as follows:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the TEN bit in USART_CTL0.
7. Wait for the TBE being asserted.
8. Write the data to the USART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 18-3. USART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the USART or entering the power saving mode. This bit can be cleared by writing 1 to TCC bit in USART_INTC register.

The break frame is sent when the SBKCMD bit is set, and SBKCMD bit is reset after the transmission.

18.3.4. USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1.
3. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the REN bit in USART_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 USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

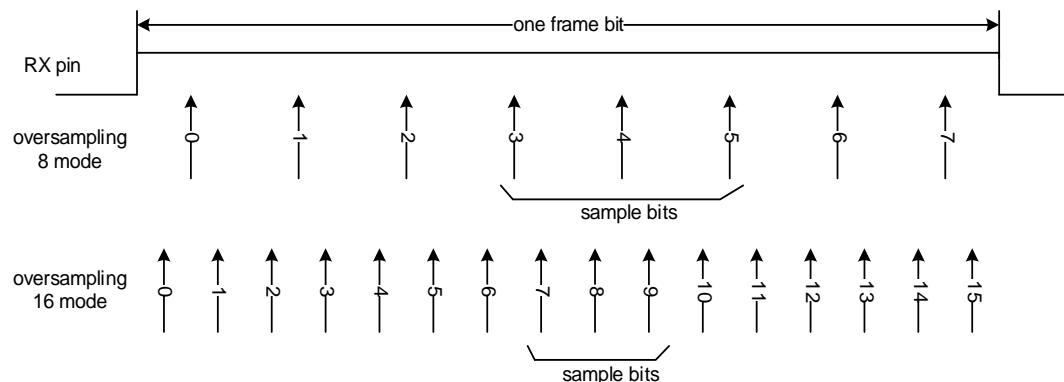
The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

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 3rd, 4th and 5th samples are used, while in the

oversampling 16 mode, the 7th, 8th, and 9th samples are used. If two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a start bit, data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt will be generated, If the receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set. If the OSB bit in USART_CTL2 register is set, the receiver gets only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 18-4. Oversampling method of a receive frame bit (OSB=0)



If the parity check function is enabled by setting the PCEN bit in the USART_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 USART_STAT register will be set. An interrupt is generated, if the PERRIE bit in USART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in USART_STAT register will be set. An interrupt is generated, If the receive DMA is enabled and the ERRIE bit in USART_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 USART_STAT register will be set. An interrupt is generated, if the receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

The RBNE, NERR, PERR, FERR and ORERR flags are always set at the same time in a reception. If the receive DMA is not enabled, software can check NERR, PERR, FERR and ORERR flags when serving the RBNE interrupt.

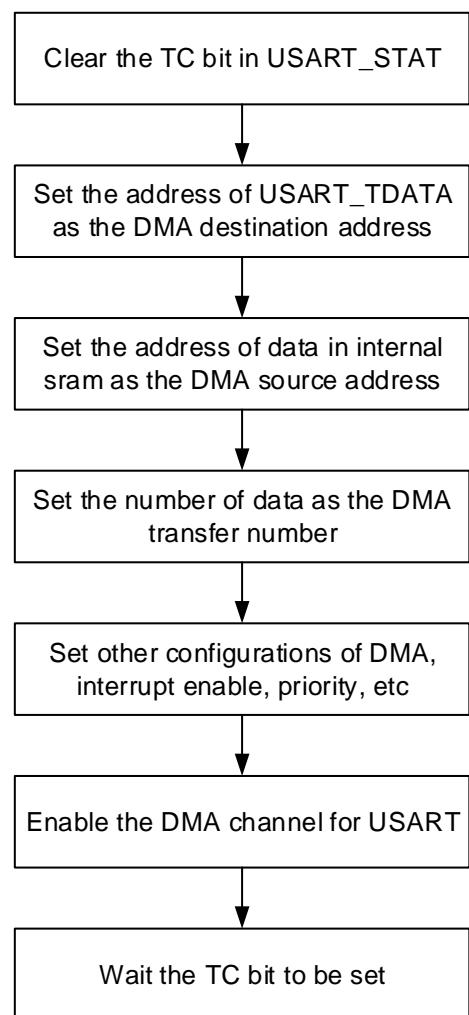
18.3.5. Use DMA for data buffer access

To reduce the burden of the processor, DMA can be used to access the transmitting and receiving data buffer. The DENT bit in USART_CTL2 is used to enable the DMA transmission, and the DENR bit in USART_CTL2 is used to enable the DMA reception.

When DMA is used for USART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the USART. The configuration step are shown in [Figure 18-5](#).

Configuration step when using DMA for USART transmission.

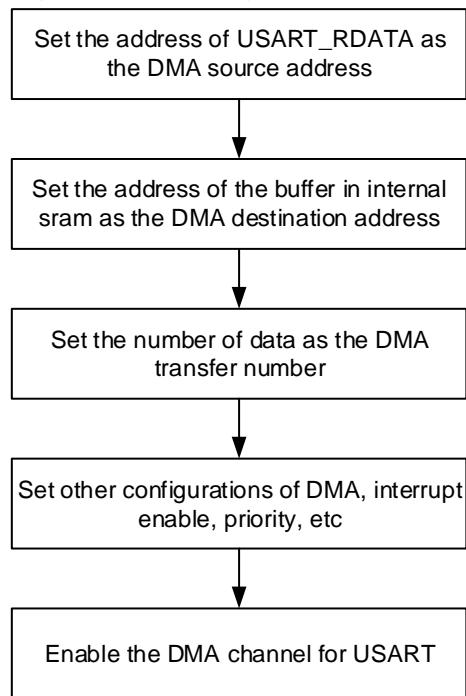
Figure 18-5. Configuration step when using DMA for USART transmission



After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in [Figure 18-6. Configuration step when using DMA for USART reception](#). If the ERRIE bit in USART_CTL2 is set, interrupts can be generated by the Error status bits (FERR, ORERR and NERR) in USART_STAT.

Figure 18-6. Configuration step when using DMA for USART reception

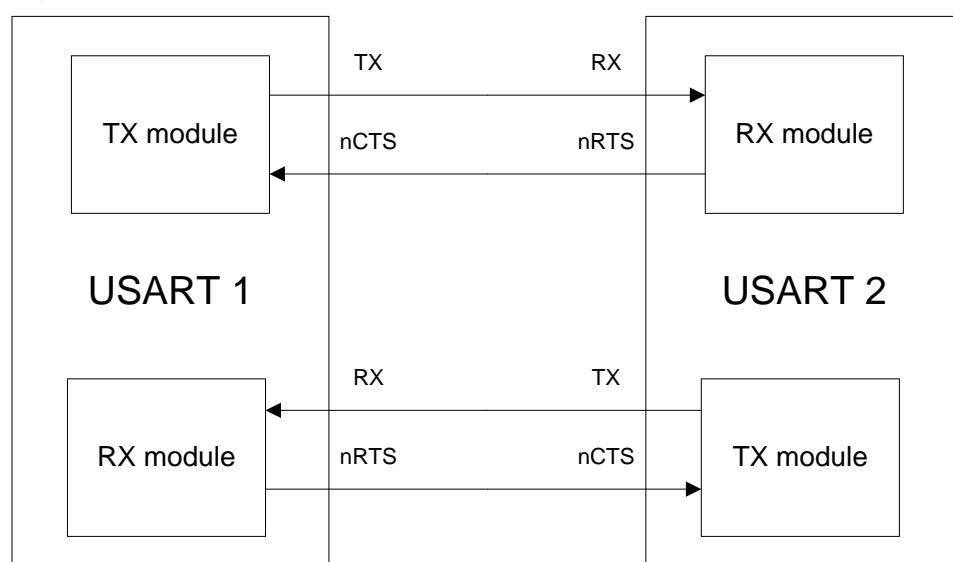


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

18.3.6. Hardware flow control

The hardware flow control function is realized by the nCTS and nRTS pins. The RTS flow control is enabled by writing ‘1’ to the RTSEN bit in USART_CTL2 and the CTS flow control is enabled by writing ‘1’ to the CTSEN bit in USART_CTL2.

Figure 18-7. Hardware flow control between two USARTs



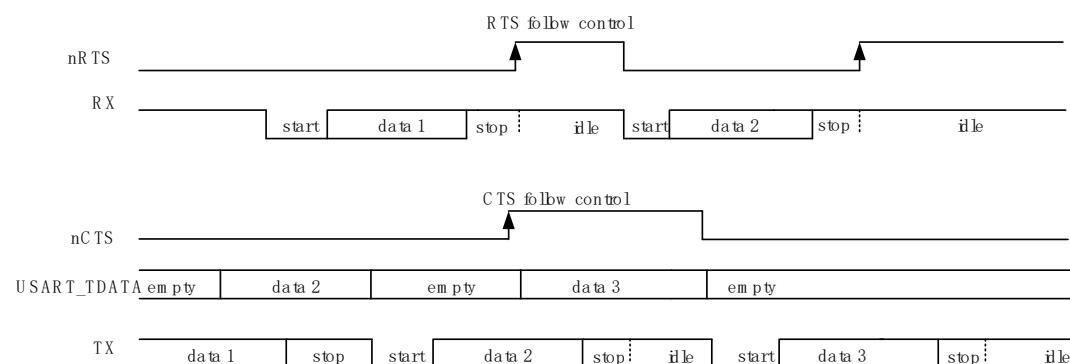
RTS flow control

The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT is ‘0’ and the nCTS signal is low, the transmitter transmits the data frame. When the nCTS signal goes high during a transmission, the transmitter stops after the current transmission is accomplished.

Figure 18-8. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the USART_CTL2 control register, allows the user to activate the external transceiver control, through the DE (Driver Enable) signal. The assertion time, which is programmed using the DEA [4:0] bits field in the USART_CTL0 control register, is the time between the activation of the DE signal and the beginning of the START bit. The de-assertion time, which is programmed using the DED [4:0] bits field in the USART_CTL0 control register, is the time between the end of the last stop bit and the de-activation of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CTL2 control register.

18.3.7. Multi-processor communication

In multiprocessor communication, several USARTs 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 USART module into a mute mode by writing 1 to the MMCMD bit in USART_CMD register.

If a USART is in mute mode, all of the receive status bits cannot be set. The USART 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 it is woken up by an idle frame, the IDLEF bit in USART_STAT will not be set.

When the WM bit of in USART_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 or 7 bits, which are configured by the ADDM bit of the USART_CTL1 register, of an address frame is the same as the ADDR bits in the USART_CTL1 register, the hardware will clear the RWU bit and exits the mute mode. The RBNE bit will be set when the frame that wakes up the USART. The status bits are available in the USART_STAT register. If the LSB 4/7 bits of an address frame defers from the ADDR bits in the USART_CTL1 register, the hardware sets the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

If the PCEN bit in USART_CTL0 is set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address bit. If the ADDM bit is set and the receive frame is a 7bit data, the LSB 6 bits will be compared with ADDR[5:0]. If the ADDM bit is set and the receive frame is a 9bit data, the LSB 8 bits will be compared with ADDR[7:0].

18.3.8. LIN mode

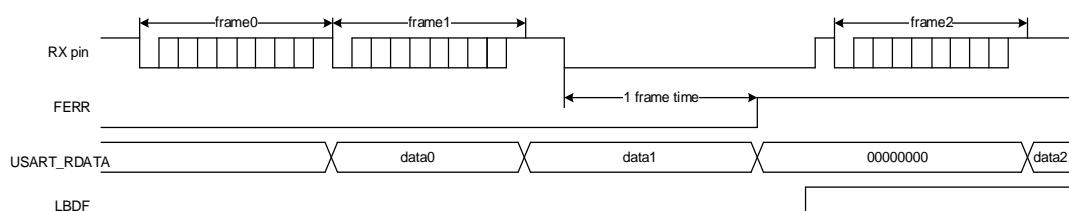
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. The data bits length can only be 8. And the break frame is 13-bit '0', followed by 1 stop bit.

The break detection function is totally independent of the normal USART 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 configuring LBLEN in USART_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 LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

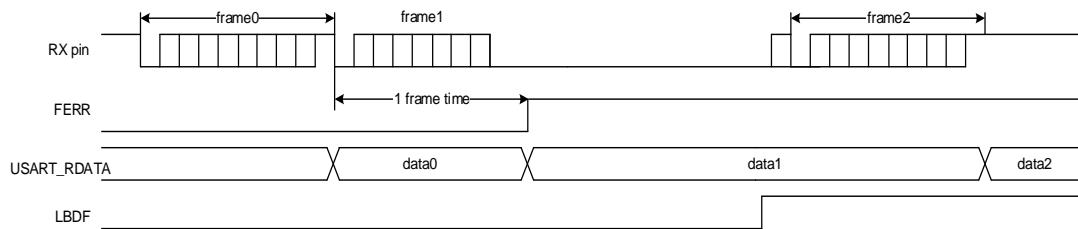
As shown in [Figure 18-9. Break frame occurs during idle state](#), if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 18-9. Break frame occurs during idle state



As shown in [Figure 18-10. 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 18-10. Break frame occurs during a frame



18.3.9. Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The clock output is also not activated during idle and break frame sending. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART Synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN=0).

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 18-11. Example of USART in synchronous mode

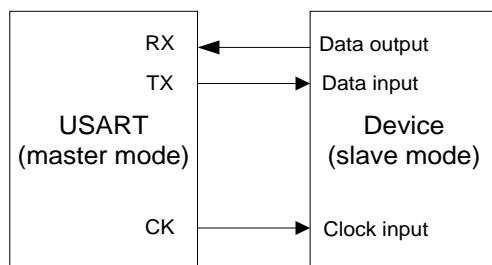
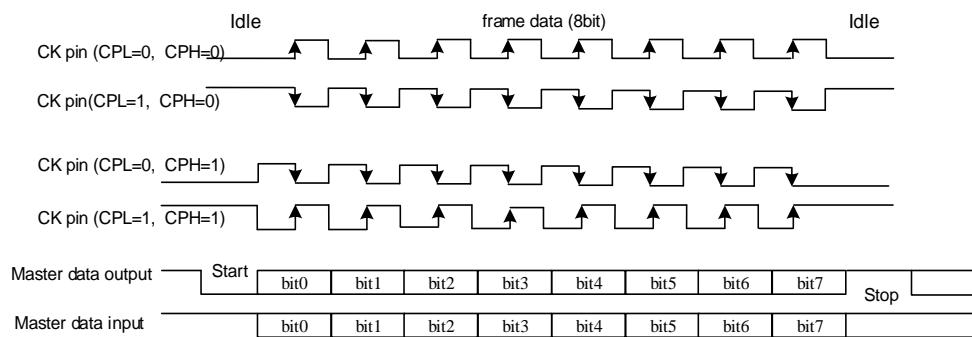
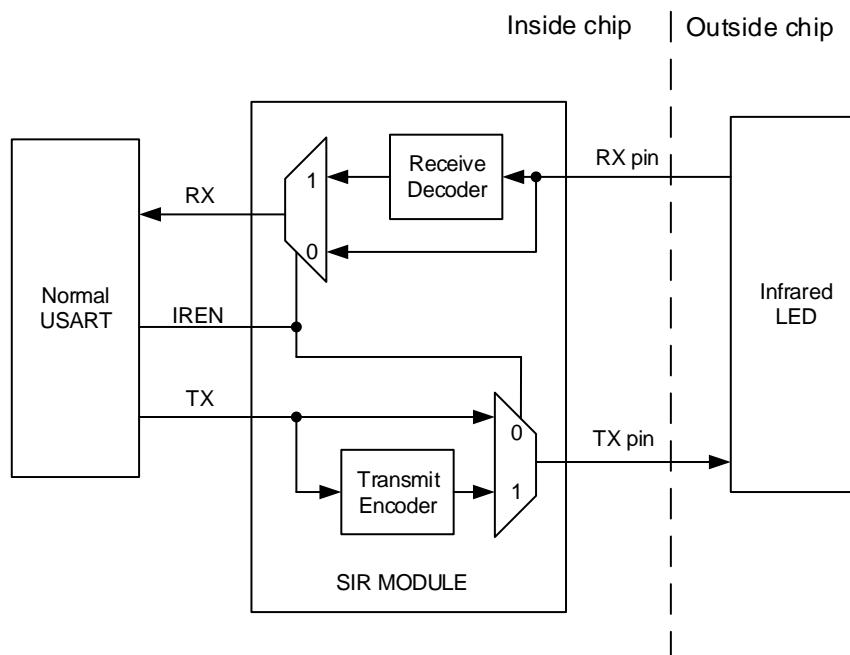


Figure 18-12. 8-bit format USART synchronous waveform (CLEN=1)


18.3.10. IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in USART_CTL2. The LMEN, STB[1:0], CKEN bits in USART_CTL1 and HDEN, SCEN bits in USART_CTL2 should be cleared in IrDA mode.

In IrDA mode, the USART 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 puts the demodulated data frame to the USART receiver. The baud rate should not be larger than 115200 for the encoder.

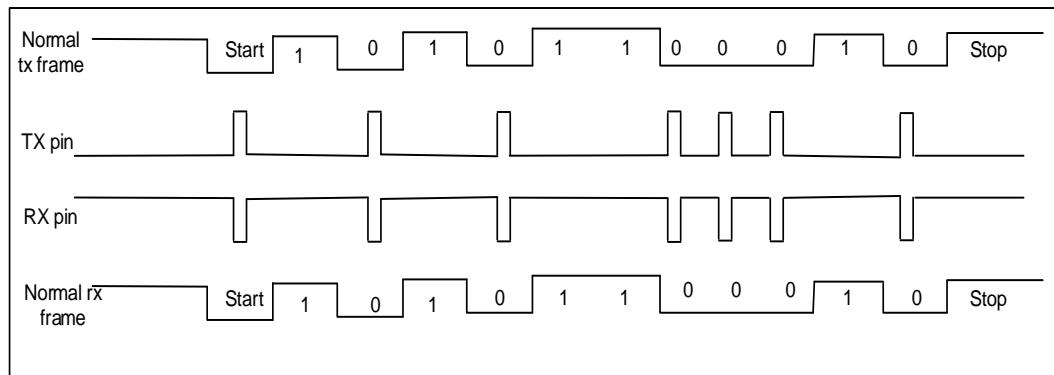
Figure 18-13. IrDA SIR ENDEC module


In IrDA mode, the polarity of the TX and RX pins 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) represents 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 of 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 18-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The division ratio is configured by the PSC[7:0] bits in USART_GP register. 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.

18.3.11. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in USART_CTL2. The LMEN, CKEN bits in USART_CTL1 and SCEN, IREN bits in USART_CTL2 should be cleared in half-duplex communication mode.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally. The TX pin should be configured as IO pin. The conflicts should be controlled by the software. When the TEN bit is set, the data in the data register will be sent.

18.3.12. Smartcard (ISO7816-3) mode

The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character ($T=0$) mode and the block ($T=1$) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

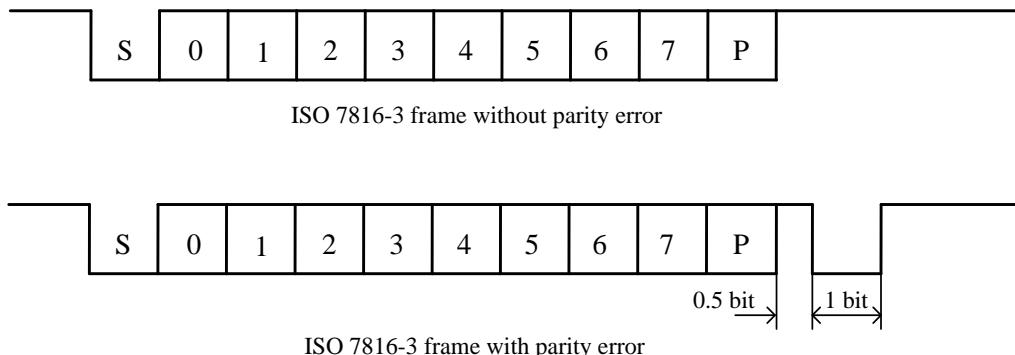
A clock is provided to the smartcard if the CKEN bit is set. The clock can be divided for other use.

The frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits.

The smartcard mode is a half-duplex communication protocol. When connected to a

smartcard, the TX pin must be configured as open drain mode, and drives a bidirectional line that is also driven by the smartcard.

Figure 18-15. ISO7816-3 frame format



Character (T=0) mode

Compared to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In Smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2.5 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE/receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T=1) mode

In block (T=1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_RT register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn't send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE=0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL (0xFF) may be programmed. The real value will be programmed after the reception of the third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to H state of the line and parity is even. In this case, the following control bits must be programmed: MSBF=0, DINV=0 (default values).

The inverse convention is defined as: MSB first, logical bit value 1 corresponds to an L state on the signal line and parity is even. In this case, the following control bits must be programmed: MSBF=1, DINV=1.

18.3.13. Auto baudrate detection

The USART is able to detect and automatically set the USART_BAUD register value based on the reception of one character. There are two methods which can be chosen through the

ABDM bits in the USART_CTL1 register. These methods are:

1. The USART will measure the duration of the start bit (falling edge to rising edge). In this case the receiving pattern should be any character starting with a bit at 1.
2. The USART will measure the duration of the start and of the 1st data bit. The measure is done falling edge to falling edge, ensuring a better accuracy in the case of slow signal slopes. In this case, the receiving pattern should be any character starting with 10xx bits.

18.3.14. ModBus communication

The USART offers basic support for the implementation of ModBus/RTU and ModBus/ASCII protocols by implementing an end of block detection.

In the ModBus/RTU mode, the end of one block is recognized by an idle line for more than 2 characters time. This function is implemented through the programmable timeout function.

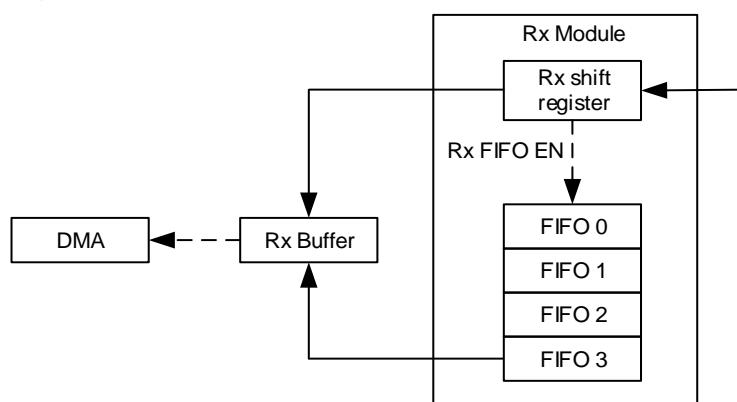
To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an interrupt will be generated, informing the software that the current block reception is completed.

In the ModBus/ASCII mode, the end of a block is recognized by a specific (CR/LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR field and activating the address match interrupt (AMIE=1). When a LF has been received or can check the CR/LF in the DMA buffer, the software will be informed.

18.3.15. Receive FIFO

The receive FIFO can be enabled by setting the RFEN bit of the USART_RFCS register to avoid the overrun error when the CPU can't serve the RBNE interrupt immediately. Up to 5 frames receive data can be stored in the receive FIFO and receive buffer. The RFFINT flag will be set when the receive FIFO is full. An interrupt is generated if the RFFIE bit is set.

Figure 18-16. USART Receive FIFO structure



If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR/NERR/FERR/EBF/ABDE/ABDF flags should be cleared before reading a receive data out.

18.3.16. Wakeup from Deep-sleep mode

The USART is able to wake up the MCU from Deep-sleep mode by the standard RBNE interrupt or the WUM interrupt.

The UESM bit must be set and the USART clock must be set to IRC16M or LXTAL (refer to the reset and clock unit RCU section).

When using the standard RBNE interrupt, the RBNEIE bit must be set before entering Deep-sleep mode.

When using the WUIE interrupt, the source of WUIE interrupt may be selected through the WUM bit fields.

DMA must be disabled before entering Deep-sleep mode. Before entering Deep-sleep mode, software must check that the USART is not performing a transfer, by checking the BSY flag in the USART_STAT register. The REA bit must be checked to ensure the USART is actually enabled.

When the wakeup event is detected, the WUF flag is set by hardware and a wakeup interrupt is generated if the WUIE bit is set, independently of whether the MCU is in stop or active mode.

18.3.17. USART interrupts

The USART interrupt events and flags are listed in [Table 18-3. USART interrupt requests](#).

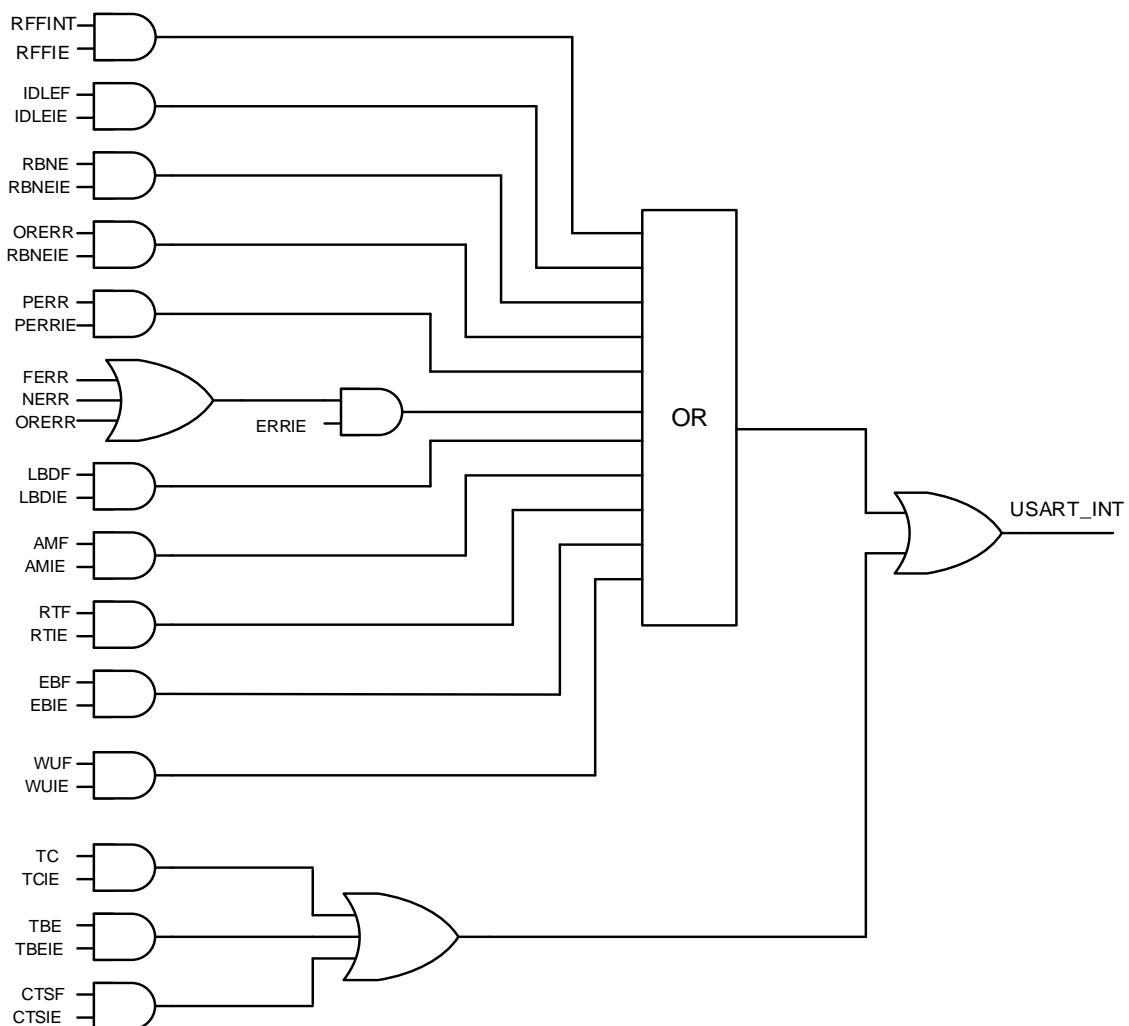
Table 18-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty	TBE	TBEIE
CTS flag	CTSF	CTSIE
Transmission complete	TC	TCIE
Received data ready to be read	RBNE	RBNEIE
Overrun error detected	ORERR	
Receive FIFO full	RFFINT	RFFIE
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception Errors (Noise flag, NERR or ORERR or FERR)	NERR or ORERR or FERR	ERRIE

Interrupt event	Event flag	Enable Control bit
overrun error, framing error) in DMA reception		
Character match	AMF	AMIE
Receiver timeout error	RTF	RTIE
End of Block	EBF	EBIE
Wakeup from Deep-sleep mode	WUF	WUIE

All of the interrupt events are ORed together before being sent to the interrupt controller, so the USART 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 18-17. USART interrupt mapping diagram



18.4. Register definition

USART0 access secure base address: 0x5000 4800
 USART0 access non-secure base address: 0x4000 4800
 USART1 access secure base address: 0x5000 4400
 USART1 access non-secure base address: 0x4000 4400
 USART2 access secure base address: 0x5001 1000
 USART2 access non-secure base address: 0x4001 1000

18.4.1. Control register 0 (USART_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				EBIE	RTIE	DEA[4:0]				DED[4:0]					
				rw	rw			rw					rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
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	EBIE	End of Block interrupt enable 0: End of Block interrupt is disabled 1: End of Block interrupt is enabled This bit is reserved in USART1.
26	RTIE	Receiver timeout interrupt enable 0: Receiver timeout interrupt is disabled 1: Receiver timeout interrupt is enabled This bit is reserved in USART1.
25:21	DEA[4:0]	Driver Enable assertion time These bits are used to define the time between the activation of the DE (Driver Enable) signal and the beginning of the start bit. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
20:16	DED[4:0]	Driver Enable de-assertion time These bits are used to define the time between the end of the last stop bit, in a transmitted message, and the de-activation of the DE (Driver Enable) signal. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the

		OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
15	OVSMOD	Oversample mode 0: Oversampling by 16 1: Oversampling by 8 This bit must be kept cleared in LIN, IrDA and smartcard modes. This bit field cannot be written when the USART is enabled (UEN=1).
14	AMIE	ADDR match interrupt enable 0: ADDR match interrupt is disabled 1: ADDR match interrupt is enabled
13	MEN	Mute mode enable 0: Mute mode disabled 1: Mute mode enabled
12	WL	Word length 0: 8 Data bits 1: 9 Data bits This bit field cannot be written when the USART is enabled (UEN=1).
11	WM	Wakeup method in mute mode 0: Idle Line 1: Address Mark This bit field cannot be written when the USART is enabled (UEN=1).
10	PCEN	Parity control enable 0: Parity control disabled 1: Parity control enabled This bit field cannot be written when the USART is enabled (UEN=1).
9	PM	Parity mode 0: Even parity 1: Odd parity This bit field cannot be written when the USART is enabled (UEN=1).
8	PERRIE	Parity error interrupt enable 0: Parity error interrupt is disabled 1: An interrupt will occur whenever the PERR bit is set in USART_STAT.
7	TBEIE	Transmitter register empty interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the TBE bit is set in USART_STAT
6	TCIE	Transmission complete interrupt enable If this bit is set, an interrupt occurs when the TC bit in USART_STAT is set. 0: Transmission complete interrupt is disabled

		1: Transmission complete interrupt is enabled
5	RBNEIE	Read data buffer not empty interrupt and overrun error interrupt enable 0: Read data register not empty interrupt and overrun error interrupt disabled 1: An interrupt will occur whenever the ORERR bit is set or the RBNE bit is set in USART_STAT.
4	IDLEIE	IDLE line detected interrupt enable 0: IDLE line detected interrupt disabled 1: An interrupt will occur whenever the IDLEF bit is set in USART_STAT.
3	TEN	Transmitter enable 0: Transmitter is disabled 1: Transmitter is enabled
2	REN	Receiver enable 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit
1	UESM	USART enable in Deep-sleep mode 0: USART not able to wake up the MCU from Deep-sleep mode. 1: USART able to wake up the MCU from Deep-sleep mode. Providing that the clock source for the USART must be IRC16M or LXTAL. This bit is reserved in USART1.
0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

18.4.2. Control register 1 (USART_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
ADDR[7:0]								RTEN	ABDM[1:0]	ABDEN	MSBF	DINV	TINV	RINV	
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	Reserved	LBDIE	LBLEN	ADDM	Reserved			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

Bits	Fields	Descriptions
31:24	ADDR[7:0]	Address of the USART terminal These bits give the address of the USART terminal. In multiprocessor communication during mute mode or Deep-sleep mode, this is

used for wakeup with address mark detection. The received frame, the MSB of which is equal to 1, will be compared to these bits. When the ADDM bit is reset, only the ADDR[3:0] bits are used to compare.

In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR[7:0] value and AMF flag is set on matching.

This bit field cannot be written when both reception (REN=1) and USART (UEN=1) are enabled.

23	RTEN	Receiver timeout enable 0: Receiver timeout function disabled 1: Receiver timeout function enabled This bit is reserved in USART1.
22:21	ABDM[1:0]	Auto baud rate mode 00: Falling edge to rising edge measurement (measurement of the start bit) 01: Falling edge to falling edge measurement (the received frame must be in a Start 10xxxxxx frame format) 10: Reserved. 11: Reserved This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
20	ABDEN	Auto baud rate enable 0: Auto baud rate detection is disabled 1: Auto baud rate detection is enabled This bit is reserved in USART1.
19	MSBF	Most significant bit first 0: Data is transmitted/received with the LSB first 1: Data is transmitted/received with the MSB first This bit field cannot be written when the USART is enabled (UEN=1).
18	DINV	Data bit level inversion 0: Data bit signal values are not inverted 1: Data bit signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).
17	TINV	TX pin level inversion 0: TX pin signal values are not inverted 1: TX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).
16	RINV	RX pin level inversion 0: RX pin signal values are not inverted 1: RX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).

15	STRP	Swap TX/RX pins
		0: The TX and RX pins functions are not swapped 1: The TX and RX pins functions are swapped This bit field cannot be written when the USART is enabled (UEN=1).
14	LMEN	LIN mode enable
		0: LIN mode disabled 1: LIN mode enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
13:12	STB[1:0]	STOP bits length
		00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit This bit field cannot be written when the USART is enabled (UEN=1).
11	CKEN	CK pin enable
		0: CK pin disabled 1: CK pin enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
10	CPL	Clock polarity
		0: Steady low value on CK pin outside transmission window in synchronous mode 1: Steady high value on CK pin outside transmission window in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1).
9	CPH	Clock phase
		0: The first clock transition is the first data capture edge in synchronous mode 1: The second clock transition is the first data capture edge in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1).
8	CLEN	CK length
		0: The clock pulse of the last data bit (MSB) is not output to the CK pin in synchronous mode 1: The clock pulse of the last data bit (MSB) is output to the CK pin in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1)
7	Reserved	Must be kept at reset value
6	LBDIE	LIN break detection interrupt enable
		0: LIN break detection interrupt is disabled 1: An interrupt will occur whenever the LBDF bit is set in USART_STAT

This bit is reserved in USART1.

5	LBLEN	LIN break frame length 0: 10 bit break detection 1: 11 bit break detection This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
4	ADDM	Address detection mode This bit is used to select between 4-bit address detection and full-bit address detection. 0: 4-bit address detection 1: full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR[5:0], ADDR[6:0] and ADDR[7:0]) respectively This bit field cannot be written when the USART is enabled (UEN=1).
3:0	Reserved	Must be kept at reset value

18.4.3. Control register 2 (USART_CTL2)

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						WUIE	WUM[1:0]		SCRTNUM[2:0]			Reserved			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	WUIE	Wakeup from Deep-sleep mode interrupt enable 0: Wakeup from Deep-sleep mode interrupt is disabled 1: Wakeup from Deep-sleep mode interrupt is enabled This bit is reserved in USART1.
21:20	WUM[1:0]	Wakeup mode from Deep-sleep mode These bits are used to specify the event which activates the WUF (Wakeup from Deep-sleep mode flag) in the USART_STAT register. 00: WUF active on address match, which is defined by ADDR and ADDM 01: Reserved 10: WUF active on Start bit

11: WUF active on RBNE

This bit field cannot be written when the USART is enabled (UEN=1).

This bit is reserved in USART1.

19:17	SCRTNUM[2:0]	<p>Smartcard auto-retry number</p> <p>In smartcard mode, these bits specify the number of retries in transmission and reception.</p> <p>In transmission mode, a transmission error (FERR bit set) will occur after this number of automatic retransmission retries.</p> <p>In reception mode, reception error (RBNE and PERR bits set) will occur after this number of erroneous reception trials.</p> <p>When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode.</p> <p>This bit field is only can be cleared to 0 when the USART is enabled (UEN=1), to stop retransmission.</p> <p>This bit is reserved in USART1.</p>
16	Reserved	Must be kept at reset value
15	DEP	<p>Driver enable polarity mode</p> <p>0: DE signal is active high</p> <p>1: DE signal is active low</p> <p>This bit field cannot be written when the USART is enabled (UEN=1)</p>
14	DEM	<p>Driver enable mode</p> <p>This bit is used to activate the external transceiver control, through the DE signal, which is output on the RTS pin.</p> <p>0: DE function is disabled</p> <p>1: DE function is enabled</p> <p>This bit field cannot be written when the USART is enabled (UEN=1).</p>
13	DDRE	<p>Disable DMA on reception error</p> <p>0: DMA is not disabled in case of reception error. The DMA request is not asserted to make sure the erroneous data is not transferred, but the next correct received data will be transferred. The RBNE is kept 0 to prevent overrun, but the corresponding error flag is set. This mode can be used in Smartcard mode</p> <p>1: DMA is disabled following a reception error. The DMA request is not asserted until the error flag is cleared. The RBNE flag and corresponding error flag will be set. The software must first disable the DMA request (DMAR = 0) or clear RBNE before clearing the error flag</p> <p>This bit field cannot be written when the USART is enabled (UEN=1).</p>
12	OVRD	<p>Overrun disable</p> <p>0: Overrun functionality is enabled. The ORERR error flag will be set when received data is not read before receiving new data, and the new data will be lost</p> <p>1: Overrun functionality is disabled. The ORERR error flag will not be set when received data is not read before receiving new data, and the new received data</p>

overwrites the previous content of the USART_RDATA register

This bit field cannot be written when the USART is enabled (UEN=1).

11	OSB	One sample bit method 0: Three sample bit method 1: One sample bit method This bit field cannot be written when the USART is enabled (UEN=1).
10	CTSIE	CTS interrupt enable 0: CTS interrupt is disabled 1: An interrupt will occur whenever the CTS bit is set in USART_STAT
9	CTSEN	CTS enable 0: CTS hardware flow control disabled 1: CTS hardware flow control enabled This bit field cannot be written when the USART is enabled (UEN=1).
8	RTSEN	RTS enable 0: RTS hardware flow control disabled 1: RTS hardware flow control enabled, data can be requested only when there is space in the receive buffer This bit field cannot be written when the USART is enabled (UEN=1).
7	DENT	DMA enable for transmission 0: DMA mode is disabled for transmission 1: DMA mode is enabled for transmission
6	DENR	DMA enable for reception 0: DMA mode is disabled for reception 1: DMA mode is enabled for reception
5	SCEN	Smartcard mode enable 0: Smartcard Mode disabled 1: Smartcard Mode enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
4	NKEN	NACK enable in Smartcard mode 0: Disable NACK transmission when parity error 1: Enable NACK transmission when parity error This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
3	HDEN	Half-duplex enable 0: Half duplex mode is disabled 1: Half duplex mode is enabled This bit field cannot be written when the USART is enabled (UEN=1).
2	IRLP	IrDA low-power

		0: Normal mode 1: Low -power mode This bit field cannot be written when the USART is enabled (UEN=1).
1	IREN	IrDA mode enable 0: IrDA disabled 1: IrDA enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1.
0	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: An interrupt will occur whenever the FERR bit or the ORERR bit or the NERR bit is set in USART_STAT in multibuffer communication

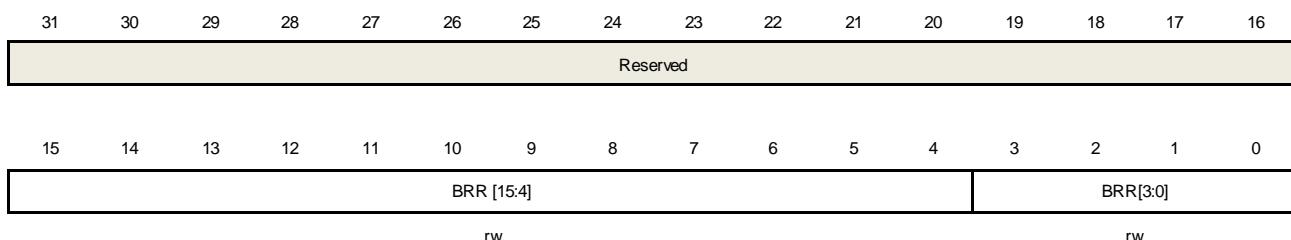
18.4.4. Baud rate generator register (USART_BAUD)

Address offset: 0x0C

Reset value: 0x0000 0000

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

This register cannot be written when the USART is enabled (UEN=1)



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:4	BRR[15:4]	Integer of baud-rate divider DIV_INT[11:0] = BRR[15:4]
3:0	BRR [3:0]	Fraction of baud-rate divider If OVSMOD = 0, USARTDIV [3:0] = BRR [3:0]; If OVSMOD = 1, USARTDIV [3:1] = BRR [2:0], BRR [3] must be reset.

18.4.5. Prescaler and guard time configuration register (USART_GP)

Address offset: 0x10

Reset value: 0x0000 0000

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

This register cannot be written when the USART is enabled (UEN=1)

This register is reserved in USART1

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	GUAT[7:0]	Guard time value in smartcard mode This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock In IrDA Low-power mode, the division factor is the prescaler value. 00000000: Reserved - do not program this value 00000001: divides the source clock by 1 00000010: divides the source clock by 2 ... In IrDA normal mode, 00000001: can be set this value only In smartcard mode, the prescaler value for dividing the system clock is stored in PSC[4:0] bits. And the bits of PSC[7:5] must be kept at reset value. The division factor is twice as the prescaler value. 00000: Reserved - do not program this value 00001: divides the source clock by 2 00010: divides the source clock by 4 00011: divides the source clock by 6 ... This bit field cannot be written when the USART is enabled (UEN=1).

18.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14

Reset value: 0x0000 0000

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

This bit is reserved in USART1

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

rw

Bits	Fields	Descriptions
31:24	BL[7:0]	<p>Block Length</p> <p>These bits specify the block length in smartcard T=1 Reception. Its value equals the number of information characters + the length of the Epilogue Field (1-LEC/2-CRC) - 1.</p> <p>This value, which must be programmed only once per received block, can be programmed after the start of the block reception (using the data from the LEN character in the Prologue Field). The block length counter is reset when TBE=0 in smartcard mode.</p> <p>In other modes, when REN=0 (receiver disabled) and/or when the EBC bit is written to 1, the Block length counter is reset.</p>
23:0	RT[23:0]	<p>Receiver timeout threshold</p> <p>These bits are used to specify receiver timeout value in terms of number of baud clocks.</p> <p>In standard mode, the RTF flag is set if no new start bit is detected for more than the RT value after the last received character.</p> <p>In smartcard mode, the CWT and BWT are implemented by this value. In this case, the timeout measurement is started from the start bit of the last received character.</p> <p>These bits can be written on the fly. The RTF flag will be set if the new value is lower than or equal to the counter. These bits must only be programmed once per received character.</p>

18.4.7. Command register (USART_CMD)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TXFCMD	RXFCMD	MMCMD	SBKCMD	ABDCMD	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value
4	TXFCMD	<p>Transmit data flush request</p> <p>Writing 1 to this bit sets the TBE flag, to discard the transmit data.</p>

This bit is reserved in USART1.

3	RXFCMD	Receive data flush command Writing 1 to this bit clears the RBNE flag to discard the received data without reading it.
2	MMCMD	Mute mode command Writing 1 to this bit makes the USART into mute mode and sets the RWU flag.
1	SBKCMD	Send break command Writing 1 to this bit sets the SBKF flag and makes the USART send a BREAK frame, as soon as the transmit machine is idle.
0	ABDCMD	Auto baudrate detection command Writing 1 to this bit issues an automatic baud rate measurement command on the next received data frame and resets the ABDF flag in the USART_STAT. This bit is reserved in USART1

18.4.8. Status register (USART_STAT)

Address offset: 0x1C

Reset value: 0x0000 00C0

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														REA	TEA
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	BSY
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABDF	ABDE	Reserved	EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	REA	Receive enable acknowledge flag This bit, which is set/reset by hardware, reflects the receive enable state of the USART core logic. 0: The USART core receiving logic has not been enabled 1: The USART core receiving logic has been enabled
21	TEA	Transmit enable acknowledge flag This bit, which is set/reset by hardware, reflects the transmit enable state of the USART core logic. 0: The USART core transmitting logic has not been enabled 1: The USART core transmitting logic has been enabled
20	WUF	Wakeup from Deep-sleep mode flag

		0: No wakeup from Deep-sleep mode 1: Wakeup from Deep-sleep mode. An interrupt is generated if WUFIE=1 in the USART_CTL2 register and the MCU is in Deep-sleep mode. This bit is set by hardware when a wakeup event, which is defined by the WUM bit field, is detected. Cleared by writing a 1 to the WUC in the USART_INTC register. This bit can also be cleared when UESM is cleared. This bit is reserved in USART1.
19	RWU	Receiver wakeup from mute mode This bit is used to indicate if the USART 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 WAKE bit in the USART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the USART_CMD register when wakeup on IDLEIE mode is selected.
18	SBF	Send break flag 0: No break character is transmitted 1: Break character will be transmitted This bit indicates that a send break character was requested. Set by software, by writing 1 to the SBKCMD bit in the USART_CMD register. Cleared by hardware during the stop bit of break transmission.
17	AMF	ADDR match flag 0: ADDR does not match the received character 1: ADDR matches the received character, An interrupt is generated if AMIE=1 in the USART_CTL0 register. Set by hardware, when the character defined by ADDR [7:0] is received. Cleared by writing 1 to the AMC in the USART_INTC register.
16	BSY	Busy flag 0: USART reception path is idle 1: USART reception path is working
15	ABDF	Auto baudrate detection flag 0: No auto baudrate detection complete 1: Auto baudrate detection complete Set by hardware when the automatic baud rate has been completed. Cleared by writing 1 to the ABDCMD in the USART_CMD register, to request a new auto baudrate detection. This bit is reserved in USART1.
14	ABDE	Auto baudrate detection error 0: No auto baudrate detection error occurred 1: Auto baudrate detection error occurred

		<p>Set by hardware if the baud rate out of range or character comparison failed.</p> <p>Cleared by software, by writing 1 to the ABDCMD bit in the USART_CMD register.</p> <p>This bit is reserved in USART1.</p>
13	Reserved	<p>Must be kept at reset value.</p>
12	EBF	<p>End of block flag.</p> <p>0: End of Block not reached.</p> <p>1: End of Block (number of characters) reached. An interrupt is generated if the EBIE=1 in the USART_CTL1 register.</p> <p>Set by hardware when the number of received bytes (from the start of the block, including the prologue) is equal or greater than BLEN + 4.</p> <p>Cleared by writing 1 to EBC bit in USART_INTC register.</p> <p>This bit is reserved in USART1.</p>
11	RTF	<p>Receiver timeout flag.</p> <p>0: Timeout value not reached.</p> <p>1: Timeout value reached without any data reception. An interrupt is generated if RTIE bit in the USART_CTL1 register is set.</p> <p>Set by hardware when the RT value, programmed in the USART_RT register has lapsed without any communication.</p> <p>Cleared by writing 1 to RTC bit in USART_INTC register.</p> <p>The timeout corresponds to the CWT or BWT timings in smartcard mode.</p> <p>This bit is reserved in USART1.</p>
10	CTS	<p>CTS level.</p> <p>This bit equals to the inverted level of the nCTS input pin.</p> <p>0: nCTS input pin is in high level.</p> <p>1: nCTS input pin is in low level.</p>
9	CTSF	<p>CTS change flag.</p> <p>0: No change occurred on the nCTS status line.</p> <p>1: A change occurred on the nCTS status line. An interrupt will occur if the CTSIE bit is set in USART_CTL2.</p> <p>Set by hardware when the nCTS input toggles.</p> <p>Cleared by writing 1 to CTSC bit in USART_INTC register.</p>
8	LBDF	<p>LIN break detected flag.</p> <p>0: LIN Break is not detected.</p> <p>1: LIN Break is detected. An interrupt will occur if the LBDIE bit is set in USART_CTL1.</p> <p>Set by hardware when the LIN break is detected.</p> <p>Cleared by writing 1 to LBDC bit in USART_INTC register.</p> <p>This bit is reserved in USART1.</p>
7	TBE	<p>Transmit data register empty.</p> <p>0: Data is not transferred to the shift register.</p>

		1: Data is transferred to the shift register. An interrupt will occur if the TBEIE bit is set in USART_CTL0 Set by hardware when the content of the USART_TDATA register has been transferred into the transmit shift register or writing 1 to TXFCMD bit of the USART_CMD register. Cleared by a write to the USART_TDATA.
6	TC	Transmission completed 0: Transmission is not completed 1: Transmission is complete. An interrupt will occur if the TCIE bit is set in USART_CTL0. Set by hardware if the transmission of a frame containing data is completed and if the TBE bit is set. Cleared by writing 1 to TCC bit in USART_INTC register.
5	RBNE	Read data buffer not empty 0: Data is not received 1: Data is received and ready to be read. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. Set by hardware when the content of the receive shift register has been transferred to the USART_RDATA. Cleared by reading the USART_RDATA or writing 1 to RXFCMD bit of the USART_CMD register.
4	IDLEF	IDLE line detected flag 0: No Idle Line is detected 1: Idle Line is detected. An interrupt will occur if the IDLEIE bit is set in USART_CTL0 Set by hardware when an Idle Line is detected. It will not be set again until the RBNE bit has been set itself. Cleared by writing 1 to IDLEC bit in USART_INTC register.
3	ORERR	Overrun error 0: No Overrun error is detected 1: Overrun error is detected. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when the word in the receive shift register is ready to be transferred into the USART_RDATA register while the RBNE bit is set. Cleared by writing 1 to OREC bit in USART_INTC register.
2	NERR	Noise error flag 0: No noise error is detected 1: Noise error is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when noise error is detected on a received frame.

Cleared by writing 1 to NEC bit in USART_INTC register.

1	FERR	Frame error flag 0: No framing error is detected 1: Frame error flag or break character is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when a de-synchronization, excessive noise or a break character is detected. This bit will be set when the maximum number of transmit attempts is reached without success (the card NACKs the data frame), when USART transmits in smartcard mode. Cleared by writing 1 to FEC bit in USART_INTC register.
0	PERR	Parity error flag 0: No parity error is detected 1: Parity error flag is detected. An interrupt will occur if the PERRIE bit is set in USART_CTL0. Set by hardware when a parity error occurs in receiver mode. Cleared by writing 1 to PEC bit in USART_INTC register.

18.4.9. Interrupt status clear register (USART_INTC)

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												WUC	Reserved	AMC	Reserved
w												w		w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EBC	RTC	Reserved	CTSC	LBDC	Reserved	TCC	Reserved	IDLEC	OREC	NEC	FEC	PEC		
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value
20	WUC	Wakeup from Deep-sleep mode clear Writing 1 to this bit clears the WUF bit in the USART_STAT register. This bit is reserved in USART1.
19:18	Reserved	Must be kept at reset value
17	AMC	ADDR match clear Writing 1 to this bit clears the AMF bit in the USART_STAT register.
16:13	Reserved	Must be kept at reset value
12	EBC	End of block clear

		Writing 1 to this bit clears the EBF bit in the USART_STAT register. This bit is reserved in USART1.
11	RTC	Receiver timeout clear Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in USART1.
10	Reserved	Must be kept at reset value
9	CTSC	CTS change clear Writing 1 to this bit clears the CTSF bit in the USART_STAT register.
8	LBDC	LIN break detected clear Writing 1 to this bit clears the LBDF flag in the USART_STAT register. This bit is reserved in USART1.
7	Reserved	Must be kept at reset value
6	TCC	Transmission complete clear Writing 1 to this bit clears the TC bit in the USART_STAT register.
5	Reserved	Must be kept at reset value
4	IDLEC	Idle line detected clear Writing 1 to this bit clears the IDLEF bit in the USART_STAT register.
3	OREC	Overrun error clear Writing 1 to this bit clears the ORERR bit in the USART_STAT register.
2	NEC	Noise detected clear Writing 1 to this bit clears the NERR bit in the USART_STAT register.
1	FEC	Frame error flag clear Writing 1 to this bit clears the FERR bit in the USART_STAT register
0	PEC	Parity error clear Writing 1 to this bit clears the PERR bit in the USART_STAT register.

18.4.10. Receive data register (USART_RDATA)

Address offset: 0x24

Reset value: Undefined

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								RDATA[8:0]							

r

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8:0	RDATA[8:0]	<p>Receive Data value</p> <p>The received data character is contained in these bits.</p> <p>The value read in the MSB (bit 7 or bit 8 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).</p>

18.4.11. Transmit data register (USART_TDATA)

Address offset: 0x28

Reset value: Undefined

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[8:0]							

rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8:0	TDATA[8:0]	<p>Transmit Data value</p> <p>The transmit data character is contained in these bits.</p> <p>The value written in the MSB (bit 7 or bit 8 depending on the data length) will be replaced by the parity, when transmitting with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).</p> <p>This register must be written only when TBE bit in USART_STAT register is set.</p>

18.4.12. USART coherence control register (USART_CHC)

Address offset: 0xC0

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	EPERR	Reserved	HCM
rc w0			rw

Bits	Fields	Descriptions
31:9	Reserved	Forced by hardware to 0.
8	EPERR	<p>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.</p> <p>0: No parity error is detected</p> <p>1: Parity error is detected.</p>
7:1	Reserved	Forced by hardware to 0.
0	HCM	<p>Hardware flow control coherence mode</p> <p>0: nRTS signal equals to the RBNE in status register</p> <p>1: nRTS signal is set when the last data bit (parity bit when pce is set) has been sampled.</p>

18.4.13. USART receive FIFO control and status register (USART_RFCS)

Address offset: 0xD0

Reset value: 0x0000 0400

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	RFFINT	Receive FIFO full interrupt flag
14:12	RFCNT[2:0]	Receive FIFO counter number
11	RFF	Receive FIFO full flag 0: Receive FIFO not full 1: Receive FIFO full
10	RFE	Receive FIFO empty flag 0: Receive FIFO not empty 1: Receive FIFO empty
9	RFFIE	Receive FIFO full interrupt enable

		0: Receive FIFO full interrupt disable 1: Receive FIFO full interrupt enable
8	RFEN	Receive FIFO enable This bit can be set when UESM = 1. 0: Receive FIFO disable 1: Receive FIFO enable
7:1	Reserved	Must be kept at reset value
0	ELNACK	Early NACK when smartcard mode is selected. The NACK pulse occurs 1/16 bit time earlier when the parity error is detected. 0:Early NACKdisable when smartcard mode is selected 1:Early NACKenable when smartcard mode is selected This bit is reserved in USART1.

19. Inter-integrated circuit interface (I2C)

19.1. 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.

19.2. 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, 1 with 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)
- Configurable SCL stretching in slave mode
- Supports DMA mode
- SMBus 3.0 and PMBus 1.3 compatible
- Optional PEC (packet error checking) generation and check
- Programmable analog and digital noise filters
- Wakeup from Deep-sleep mode on I2C0 address match
- Independent clock from PCLK

19.3. Function overview

[Figure 19-1. I2C module block diagram](#) below provides details on the internal configuration of the I2C interface.

Figure 19-1. I2C module block diagram

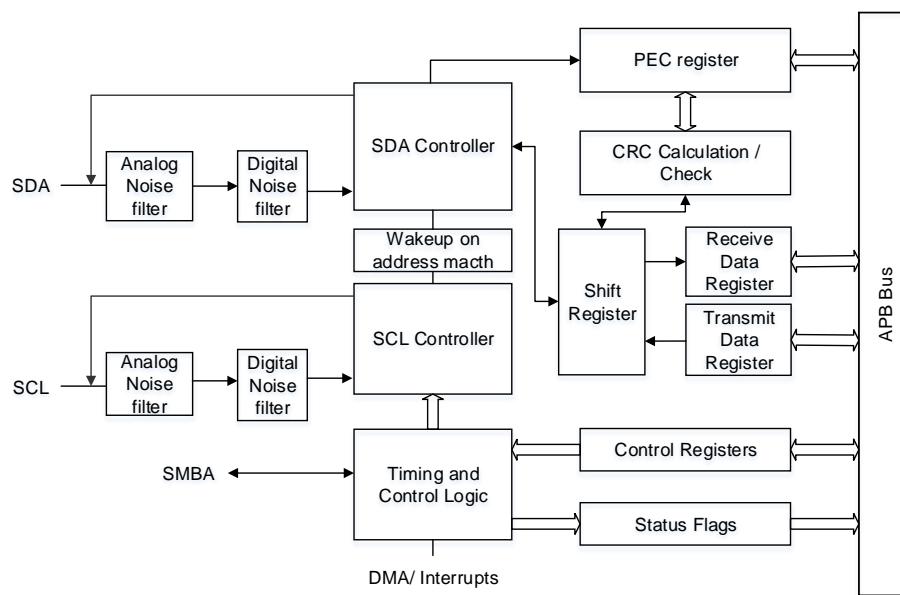


Table 19-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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

19.3.1. 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 three clock sources:

- PCLK1: APB1 clock (default value)
- IRC16M: internal 16 MHz RC
- SYSCLK: system clock

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 analog filter and digital filter.

Analog filter delay is maximum 260ns. Digital filter delay is DNF[3:0] x tI2CCLK

The period of PCLK clock t_{PCLK} match the conditions as follows:

- $t_{\text{PCLK}} < 4/3 \cdot t_{\text{SCL}}$

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} .

19.3.2. I2C communication flow

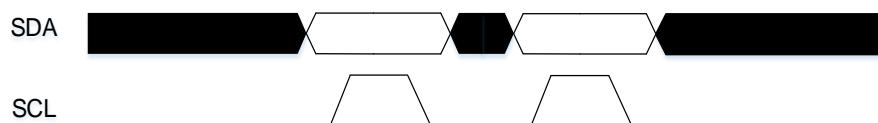
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- 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 19-2. Data validation](#)). One clock pulse is generated for each data bit transferred.

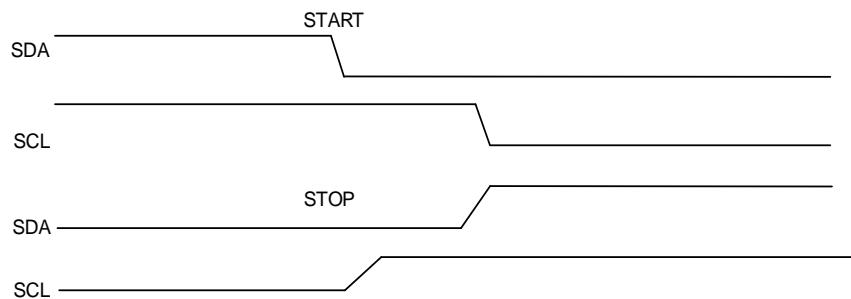
Figure 19-2. Data validation



START and STOP condition

All transactions begin with a START (S) and are terminated by a STOP (P) (see [Figure 19-3. START and STOP condition](#)). A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition.

Figure 19-3. START and STOP condition



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 condition, 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 condition 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 condition 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 condition and it's also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP condition is generated automatically by hardware. If AUTOEND=0, the STOP condition generated by software, or the master can generate a RESTART condition to start a new transfer.

Figure 19-4. I²C communication flow with 10-bit address (Master Transmit)

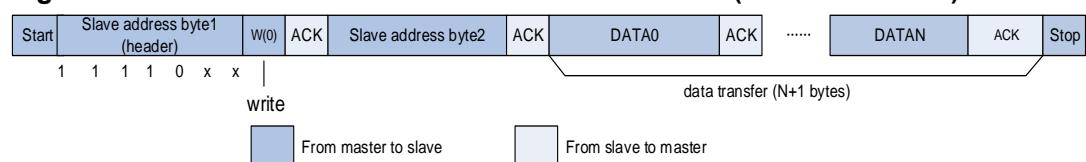
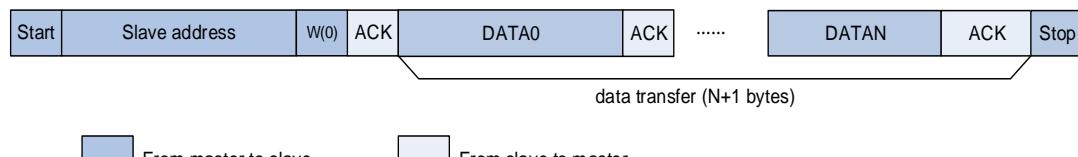
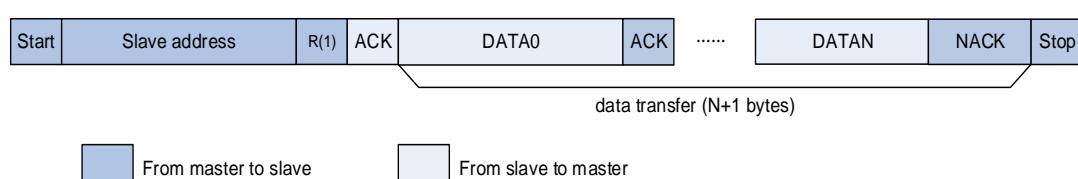
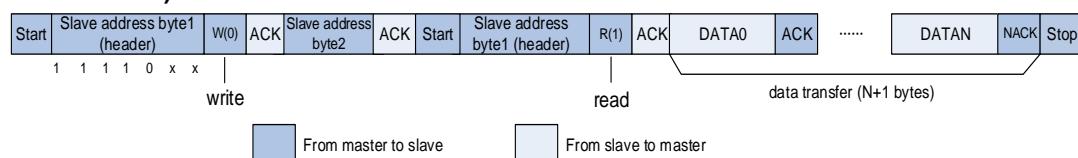
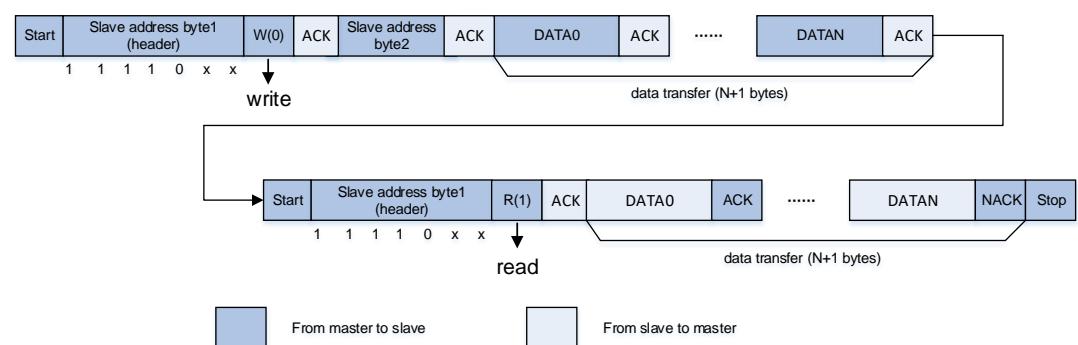


Figure 19-5. I2C communication flow with 7-bit address (Master Transmit)

Figure 19-6. 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 19-7. 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 19-8. I2C communication flow with 10-bit address \(Master Receive when HEAD10R=1\)](#).

Figure 19-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

Figure 19-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)


19.3.3. Noise filter

The noise filters must be configured before setting the I2CEN bit in I2C_CTL0 register if it is necessary. The analog noise filter is present on the SDA and SCL inputs by default. The analog filter requires the suppression of spikes with a pulse width up to 50ns in fast mode and fast mode plus. The analog filter can be disabled by setting the ANOFF bit in I2C_CTL0 register.

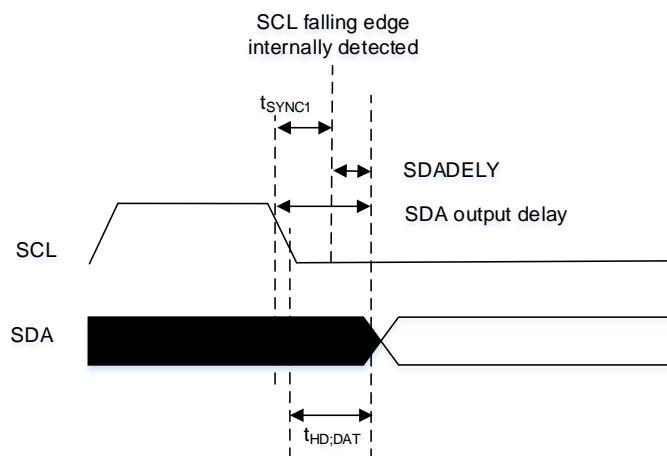
The digital filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. When the digital filter is enabled, the level of the SCL or the SDA line is internally changed only if it remains stable for more than $DNF[3:0] \times t_{I2CCLK}$. This allows to suppress spikes with a programmable length of 1 to 15 of t_{I2CCLK} .

19.3.4. I2C timings

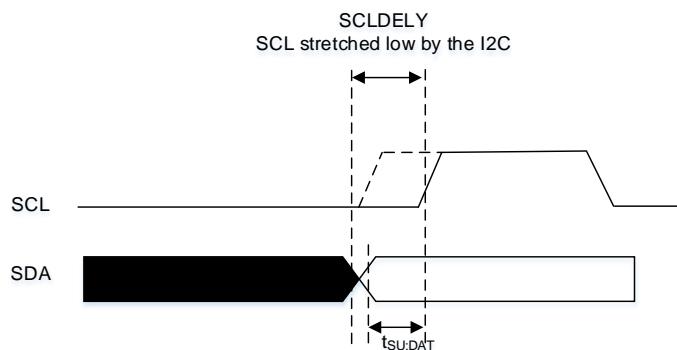
The PSC[3:0], SCLDELY[3:0] and SDADELY[3:0] bits in the I2C_TIMING register must be configured in order to guarantee a correct data hold and setup time used in I2C communication.

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 19-9. Data hold time](#).

Figure 19-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in [Figure 19-10. Data setup time](#).

Figure 19-10. Data setup time


When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDADEV} = SDADEV * t_{PSC} + t_{I2CCLK}$ where $t_{PSC} = (PSC+1) * t_{I2CCLK}$. t_{SDADEV} effects $t_{HD;DAT}$. The total delay of SDA output is $t_{SYNC1} + \{[SDADEV * (PSC+1) + 1] * t_{I2CCLK}\}$. t_{SYNC1} depends on SCL falling slope, the delay of analog filter, 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} .

SDADELY must match condition as follows:

- $SDADELY \geq \{t_r(\max) + t_{HD;DAT}(\min) - t_{AF}(\min) - [(DNF+3) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$
- $SDADELY \leq \{t_{HD;DAT}(\max) - t_{AF}(\max) - [(DNF+4) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$

Note: The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When SS = 0, after t_{SDADEV} 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 $t_{SCLDEV} = (SCLDEV+1) * t_{PSC}$. t_{SCLDEV} effects $t_{SU;DAT}$.

SCLDEV must match condition as follows:

$$SCLDEV \geq \{[t_r(\max) + t_{SU;DAT}(\min)] / [(PSC+1) * t_{I2CCLK}]\} - 1$$

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 19-2. 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	

19.3.5. Software 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:

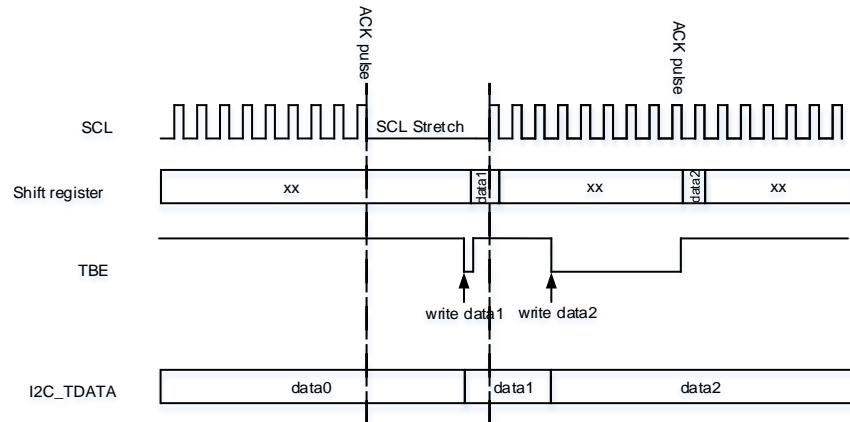
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

19.3.6. Data transfer

The data transfer is managed through data registers and shift register.

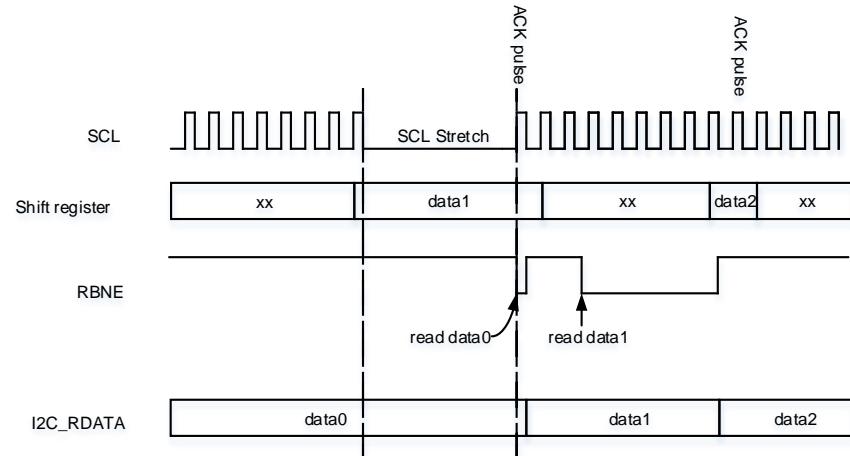
Data Transmission

If the I2C_TDATA register is not empty, that is, TBE=0, its content is moved to the shift register after the 9th SCL pulse (Acknowledge pulse). The shift register content is shifted out on SDA line. If TBE=1 means that no data is written to I2C_TDATA, SCL line is stretched low until I2C_TDATA is written. The stretch is done after the 9th SCL pulse.

Figure 19-11. Data transmission


Data Reception

When receiving data, the SDA input fills the shift register. After the 8th SCL pulse, the complete data byte is received. If RBNE=0 (I2C_RDATA register is empty), the data in the shift register is moved into I2C_RDATA register. If RBNE=1 indicates that the previous received data byte has not been read, the SCL line is stretched low until I2C_RDATA is read. The stretch is inserted between the 8th and 9th SCL pulse (before Acknowledge pulse).

Figure 19-12. Data reception


Hardware transfer management

In order to manage byte transfer and to shut down the communication in modes as is shown in [Table 19-3. Communication modes to be shut down](#), the I2C embedded a byte counter in the hardware.

Table 19-3. 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 byte counter is always used in master mode. It is disabled in slave mode by default, but it can be enabled by software by setting the SBCTL (slave byte control) bit in the I2C_CTL0 register.

The number of bytes to be transferred is programmed in the BYTENUM[7:0] bit field of the I2C_CTL1 register. If the number of bytes to be transferred (BYTENUM) is greater than 255, or if a receiver wants to control the acknowledge value of a received data byte, the reload mode must be selected by setting the RELOAD bit in the I2C_CTL1 register. In this mode, TCR flag is set when the number of bytes programmed in BYTENUM has been transferred, and an interrupt is generated if TCIE is set. Once the TCR flag is set, SCL is stretched. When BYTENUM is written to a non-zero value, TCR is cleared by software.

When the BYTENUM counter is reloaded with the last number of bytes, RELOAD bit must be cleared.

When RELOAD = 0 in master mode, the counter can be used in two modes:

- **Automatic end mode** (AUTOEND = 1 in the I2C_CTL1 register). In this mode, once the number of bytes programmed in the BYTENUM[7:0] bit field has been transferred, the master automatically sends a STOP condition. Note that when the RELOAD bit is set, the AUTOEND bit has no effect.
- **Software end mode** (AUTOEND = 0 in the I2C_CTL1 register). In this mode, once the number of bytes programmed in the BYTENUM[7:0] bit field has been transferred, the TC flag is set and an interrupt is generated if the TCIE bit is set. As long as the TC flag is set, the SCL signal is stretched. When the START or STOP bit is set in the I2C_CTL1 register, the TC flag is cleared by software. This mode must be used when the master wants to send a RESTART condition.

Table 19-4. I2C configuration

Function	SBCTL bit	RELOAD bit	AUTOEND bit
Master Tx/Rx + BYTENUM + STOP	x	0	1
Master Tx/Rx + BYTENUM + RESTART	x	0	0
Slave Tx/Rx all received bytes ACKed	0	x	x
Slave Rx with ACK control	1	1	x

19.3.7. 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.
- After SCL falling edge detection, the I2C stretches SCL low during $[(SDADELY+SCLDELY+1) \times (PSC+1) + 1] \times tI2CCLK$.

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.

- The SCL will be not stretched while the ADDSEND is set.

- In slave transmitting mode, the data should be written in the I2C_TDATA register before the first SCL pulse corresponding to its transfer occurs. 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, the data must be read from the I2C_RDATA register before the 9th SCL pulse (ACK pulse) occurred by the next data byte. 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.

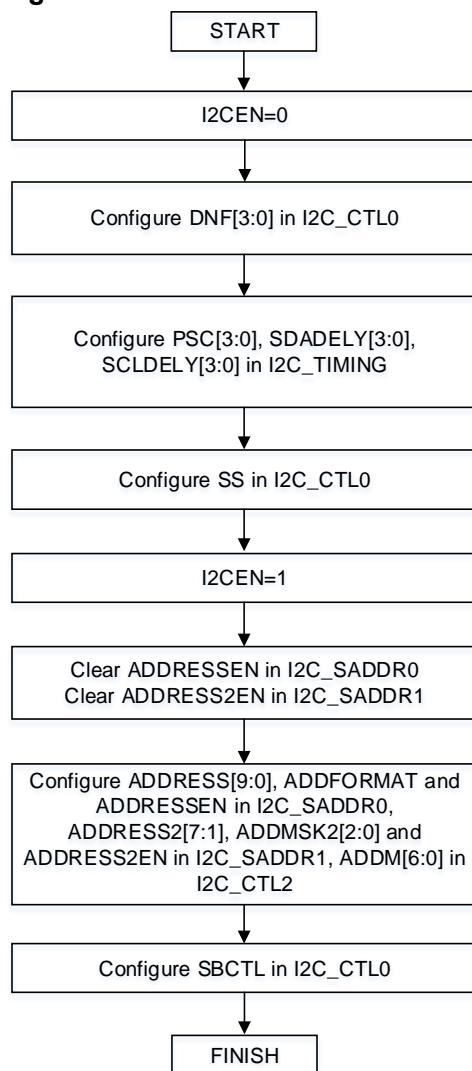
When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In order to get control of each byte, BYTENUM[7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the slave stretches the SCL low between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determined 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. The ACK or NACK is sent and the next byte can be received.

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 19-13. I2C initialization in slave mode


Slave transmitter

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 data byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the data written in the last TI event of the last transfer.
- If the data is not the one to be sent, setting the TBE bit can flush the data, then a new byte can be written in I2C_TDATA register. Then the STPDET bit should be cleared. 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 19-14. Programming model for slave transmitting when SS=0

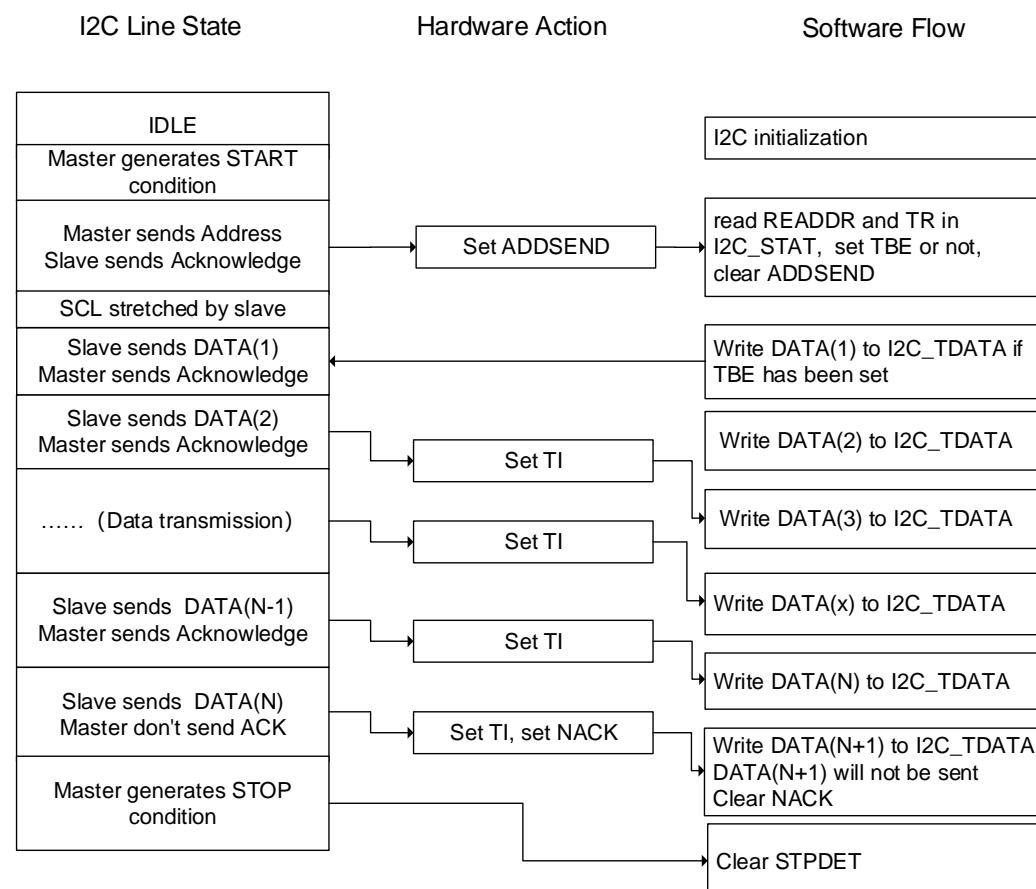
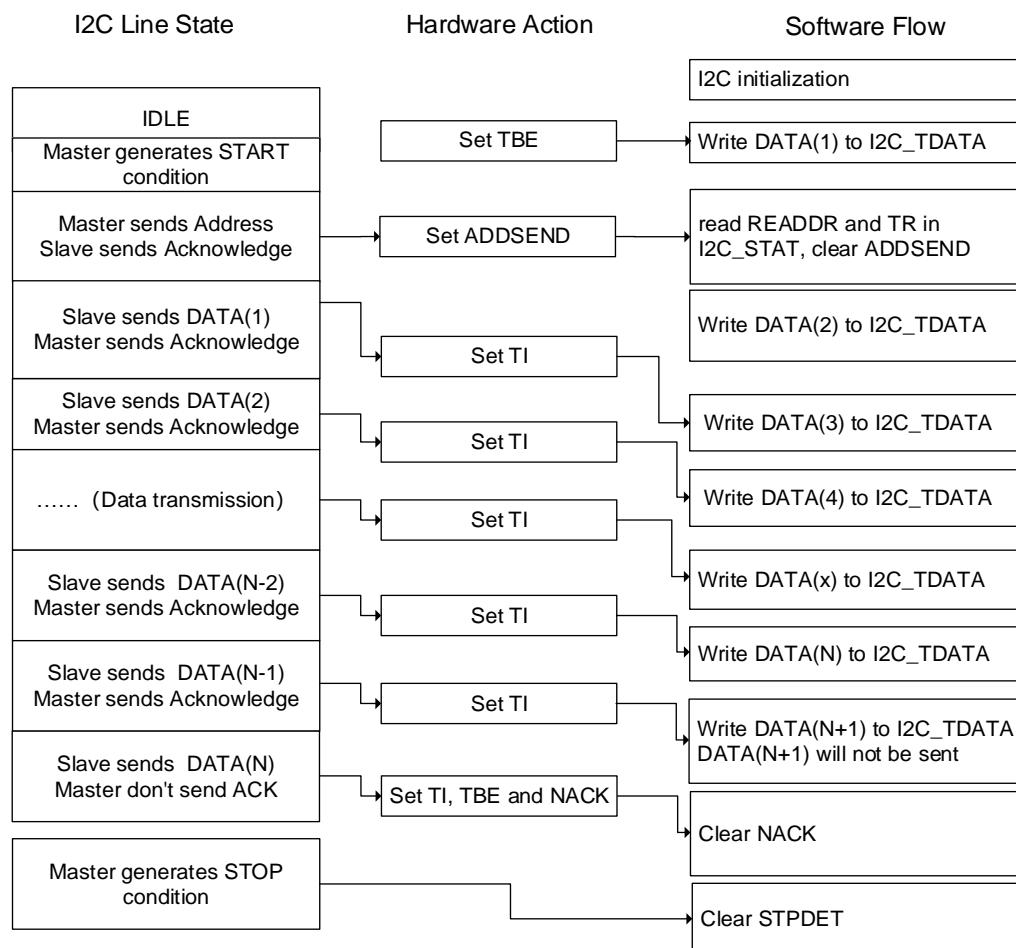


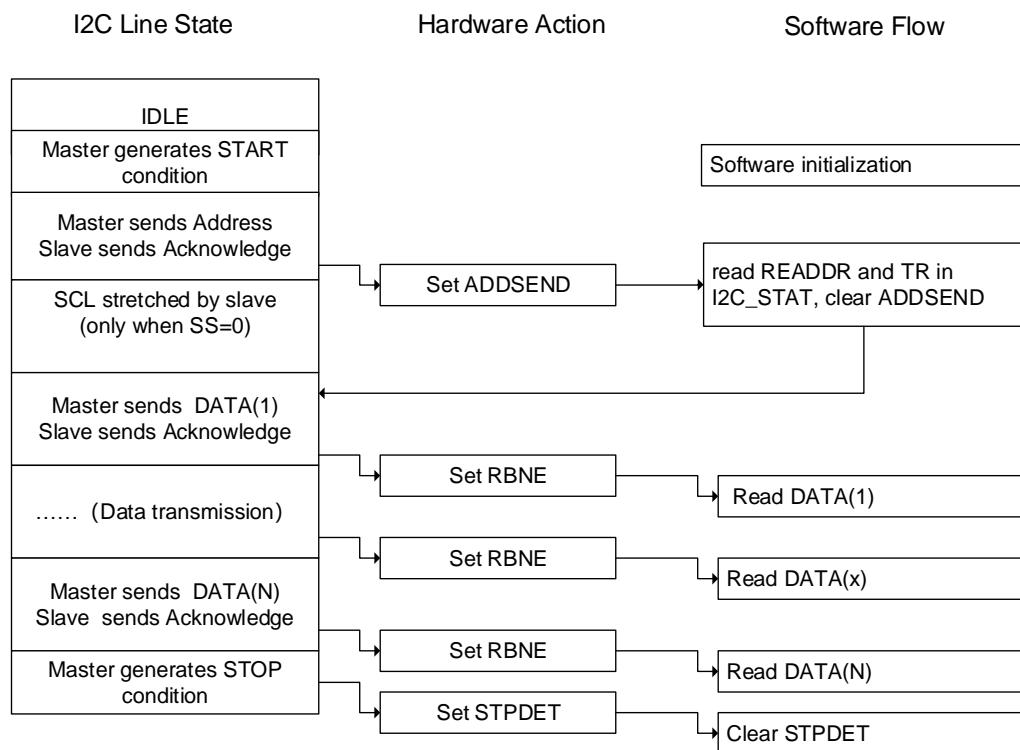
Figure 19-15. Programming model for slave transmitting when SS=1



Slave receiver

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 19-16. Programming model for slave receiving



19.3.8. 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.

For clock synchronization, the low level of the clock is counted starting from the SCL low level internal detection by the SCLL[7:0] counter, the high level of the clock is counted by the SCLH[7:0] counter, starting from the SCL high level internal detection.

The I2C detects its SCL low level after a t_{SYNC1} delay depending on the SCL falling edge, SCL input analog and digital noise filter and SCL synchronization to the I2CCLK clock. If the SCLL[7:0] value in I2C_TIMING register is reached by the SCLL[7:0] counter, the I2C will release the SCL clock.

The I2C detects its SCL high level after a t_{SYNC2} delay depending on the SCL rising edge, SCL input analog and digital noise filter and SCL synchronization to I2CCLK clock. If the SCLH[7:0] value in I2C_TIMING register is reached by the SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is: $t_{SCL} = t_{SYNC1} + t_{SYNC2} + [(SCLH[7:0]+1) + (SCLL[7:0]+1)] \times 643$

$(PSC+1) \times t_{I2CCLK}$

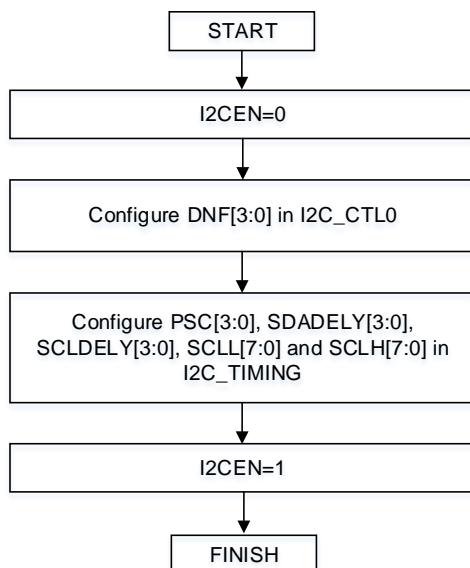
The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The delay by digital noise filter is $DNF[3:0] \times t_{I2CCLK}$.

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 condition. All the bits above should be configured before the START is set. The slave address will be sent after the START condition 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.

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 ADDSENDC 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 ADDSENDC bit must be set to clear the START bit.

Figure 19-17. I2C initialization in master mode



Master transmitter

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_CTL0 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL0 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 condition 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 condition by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART condition to start a new transfer. The TC bit is cleared when the START/STOP bit is set.
- If a NACK is received, a STOP condition 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 19-18. Programming model for master transmitting (N<=255)

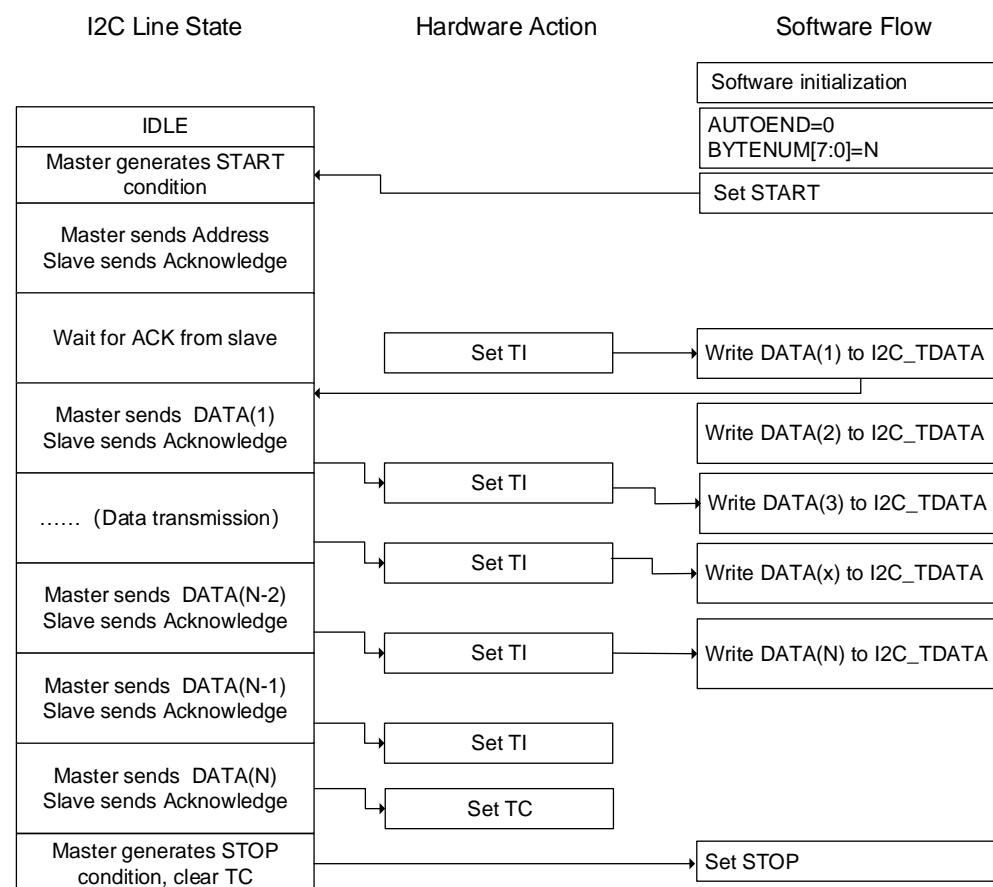
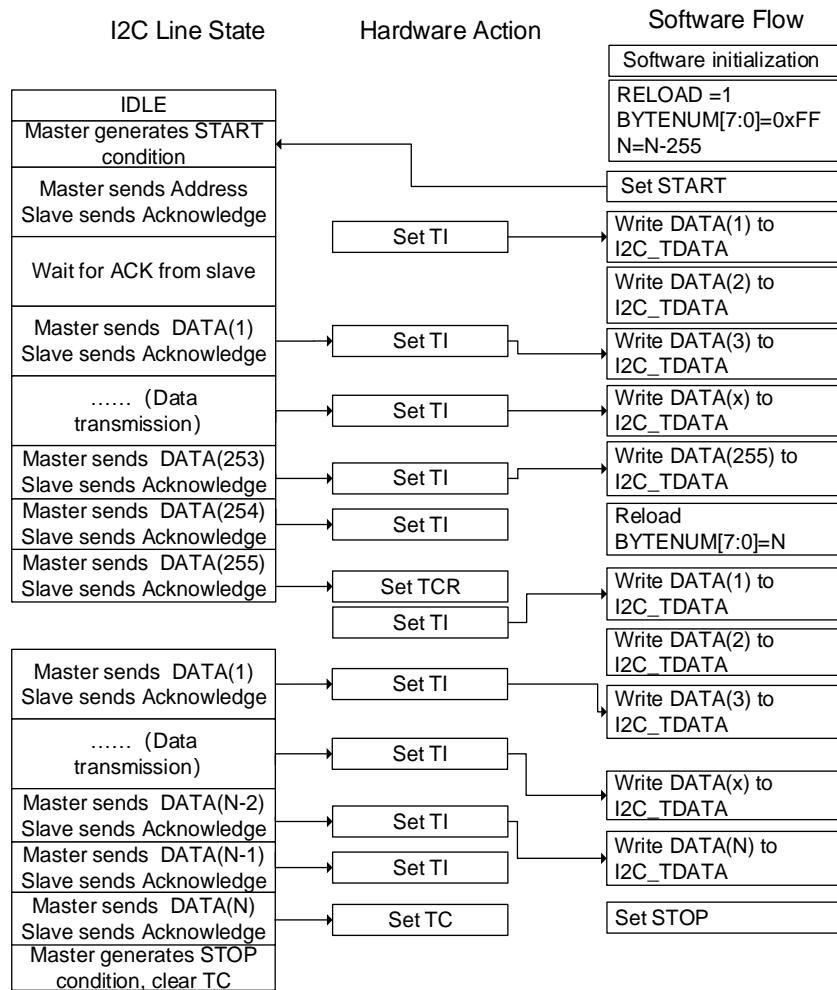


Figure 19-19. Programming model for master transmitting (N>255)


Master receiver

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_CTL0 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 condition 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 condition by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART condition to start a new transfer. The TC bit is cleared when the START bit is set.

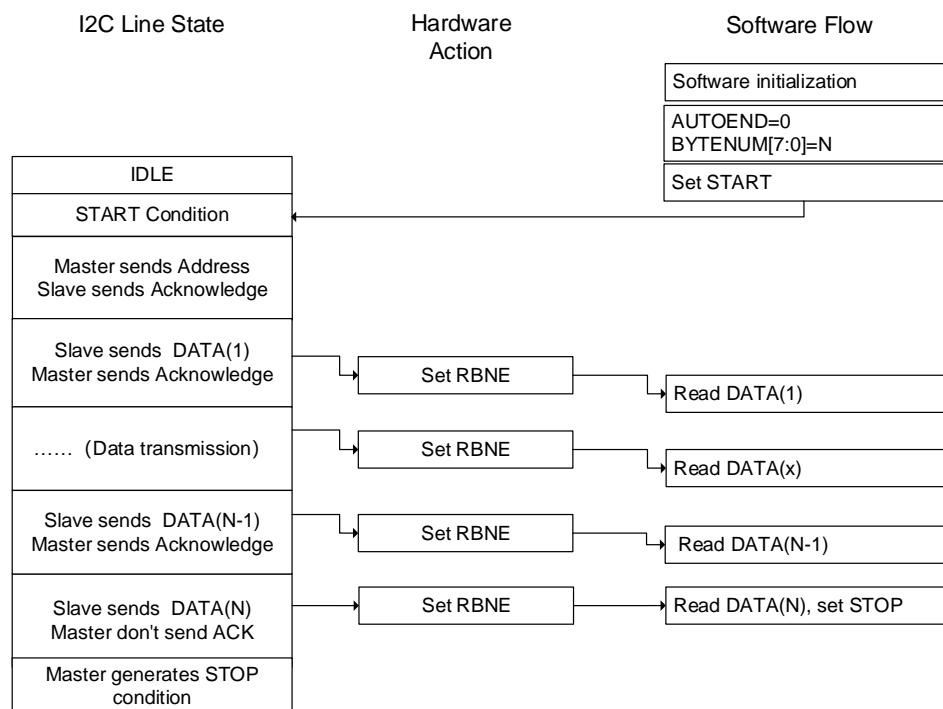
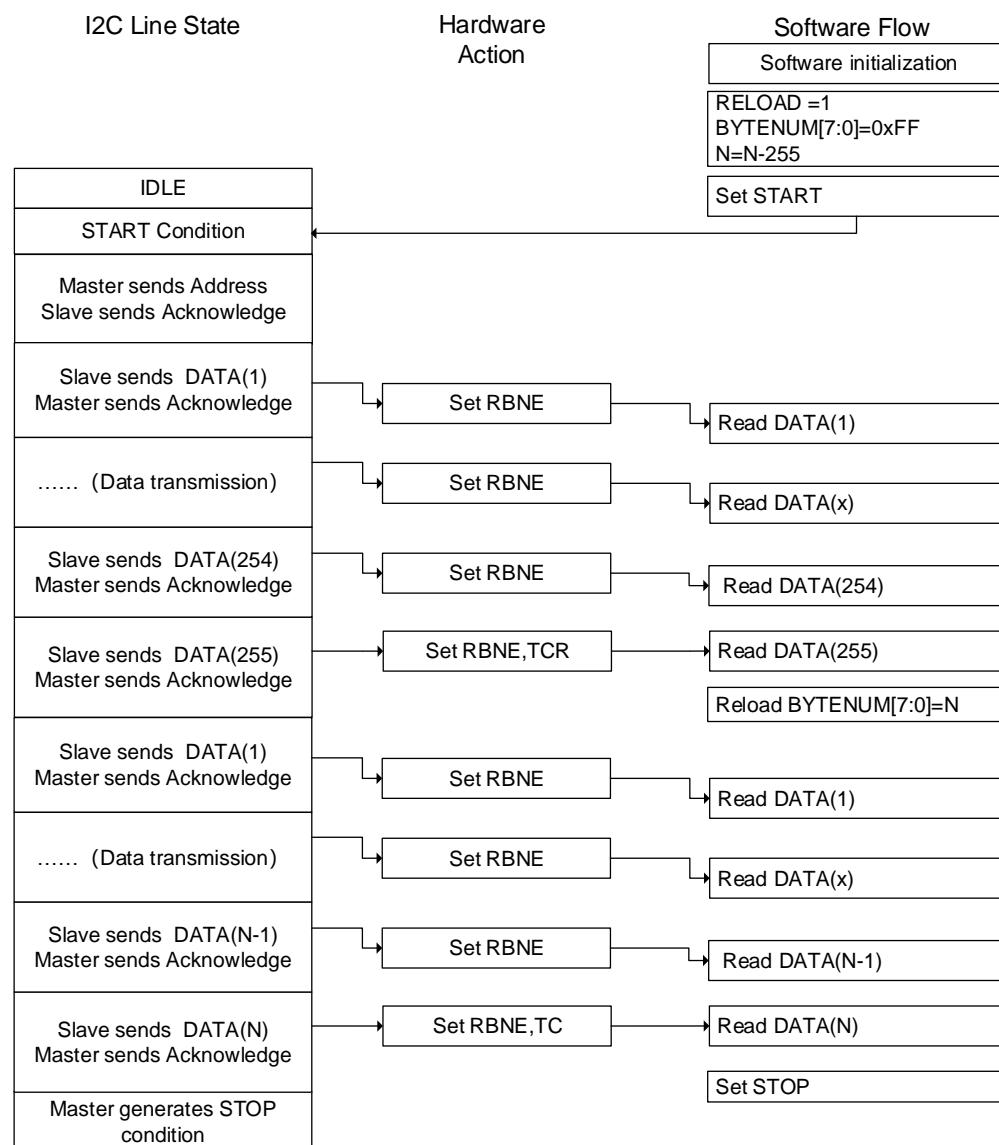
Figure 19-20. Programming model for master receiving (N<=255)


Figure 19-21. Programming model for master receiving (N>255)


19.3.9. 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 both those protocols there is a very useful distinction made between a System Host and all the other devices in the system that can have the names and functions of masters or slaves.

Received command and data acknowledge control

A SMBus receiver must be able to NACK each received command or data. In order to allow the ACK control in slave mode, 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. The host will acknowledge the SMBus Host address. When this protocol is used, the device acts as a master and the host as a slave.

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 EXTOEN 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.

- t_{TIMEOUT} check

In order to enable the $t_{TIMEOUT}$, the BUSTOA[11:0] must be programmed with the timer to check the $t_{TIMEOUT}$ parameter. To detect SCL low level timeout, the TOIDLE bit must be configured to "0". Then set TOEN in the I2C_TIMEOUT register to enable the timer. If the low level time of SCL is greater than $(BUSTOA + 1) \times 2048 \times t_{I2CCLK}$, the TIMEOUT flag is set in the I2C_STAT register.

Note: After the TOEN bit is set, the BUSTOA[11:0] and the TOIDLE bit cannot be changed.

- $t_{LOW:SEXT}$ and $t_{LOW:MEXT}$ check

By configuring a 12-bit BUSTOB timer, the $t_{LOW:SEXT}$ of the slave and the $t_{LOW:MEXT}$ of the master can be checked. Since the standard only specifies the maximum value, the same value can be chose for the both. Then enable the timer by setting the EXTOEN bit in the I2C_TIMEOUT register. If the SCL stretching time of the SMBus peripheral is longer than $(BUSTOB + 1) \times 2048 \times t_{I2CCLK}$ and within the timeout interval described in the Bus idle detection section, the TIMEOUT bit in the I2C_STAT register will be set.

Note: After the TOEN bit is set, the BUSTOB[11:0] cannot be changed.

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).

Setting the PECEN bit in the I2C_CTL0 register will enable the PEC calculation. The PEC transfer can be managed by the hardware byte counter: BYTENUM[7:0] in the I2C_CTL1 register. The PECEN bit must be configured before enabling the I2C.

Since the PEC transmission is managed by hardware byte counter, SBCTL bit must be set when connecting SMBus in slave mode. 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.

Table 19-5. SMBus with PEC configuration

Mode	SBCTL bit	RELOAD bit	AUTOEND bit	PECTRANS bit
Master Tx/Rx BYTENUM + PEC+ STOP	x	0	1	1
Master Tx/Rx BYTENUM + PEC + RESTART	x	0	0	1
Slave Tx/Rx with PEC	1	0	x	1

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. Only the device(s) which pulled SMBALERT# low will acknowledge the Alert Response Address. When SMBHAEN is 0, it is configured as a slave device, the SMBA pin is 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 is 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 ALERT line is considered high even if the external 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 peripheral supports hardware bus idle detection.

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 set TOEN in the I2C_TIMEOUT register to enable the timer. If the high level time of both SCL and SDA is greater than $(BUSTOA + 1) \times 4 \times t_{I2CCLK}$, the TIMEOUT flag is set in the I2C_STAT register.

Note: After the TOEN bit is set, the BUSTOA[11:0] bit and the TOIDLE bit cannot be changed.

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 (0b0001100) is enabled by setting the SMBALTEN bit in the I2C_CTL0 register.

19.3.10. SMBus mode

SMBus Master Transmitter and Slave Receiver

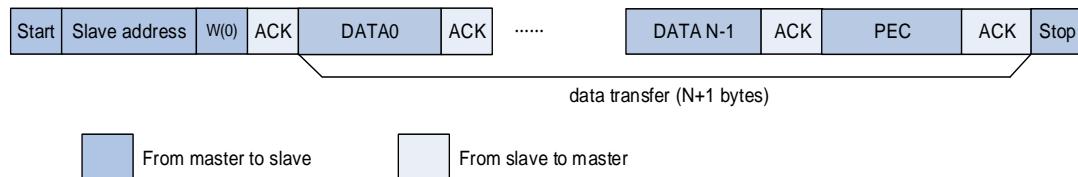
The PEC in SMBus master mode can be transmitted by setting the PECTRANS bit before setting the START bit, and the number of bytes in the BYTENUM[7:0] field must be configured. In this case, the total number of transmissions when TI interrupt occurs is BYTENUM-1. So if

BYTEENUM=0x1 and PECTRANS bit is set at the same time, the contents of the I2C_PEC register are automatically transferred. If the automatic end mode is selected (AUTOEND=1), the SMBus master automatically sends the STOP condition after the PEC byte. If the automatic end mode is not selected (AUTOEND=0), the SMBus master can send a RESTART condition after the PEC. The I2C_PEC register content will be sent after BYTEENUM -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 to enable the RELOAD mode. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTEENUM-1 data, the next received byte will be compared with the contents of the I2C_PEC register. If the comparison does not match, the NACK is automatically generated; if the comparison matches, the ACK is automatically generated, regardless of the ACK bit value. When PEC byte is received, it is also copied into the I2C_RDATA register like other data, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC 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 BYTEENUM 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 19-22. 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 chose (AUTOEND=1). Before sending a START condition on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTEENUM-1 data, the next received byte will be compared with the contents of the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP condition.

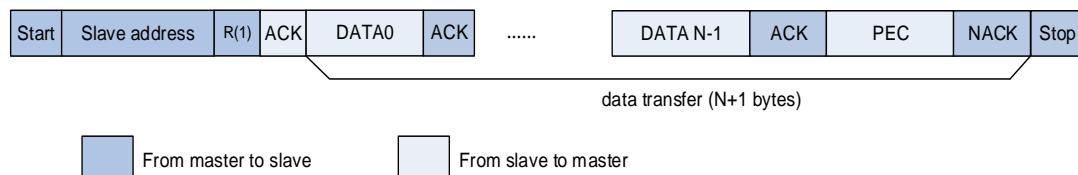
If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, the software mode (AUTOEND = 0) must be selected. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTEENUM-1 data, the next received byte will be compared with the contents of 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 BYTEENUM[7:0] bytes, SBCTL must be set. If PECTRANS bit is set, the number of bytes in BYTEENUM[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 contents of the I2C_PEC register will be transmitted automatically.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 19-23. SMBus Master Receiver and Slave Transmitter communication flow



19.3.11. Wakeup from Deep-sleep mode

When the address of I2C matches correctly, it can wake up from MCU Deep-sleep mode (APB clock is off). In order to wake up from Deep-sleep mode, WUEN bit must be set in the I2C_CTL0 register and the IRC16M must be selected as the clock source for I2CCLK. During Deep-sleep mode, the IRC16M is switched off. The I2C interface switches the IRC16M on, and stretches SCL low until IRC16M is woken up when a START is detected. Then the IRC16M 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, IRC16M 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, IRC16M will not be switched on after receiving start signal. When wakeup from Deep-sleep mode is enabled, the digital filter must be disabled and the SS bit in I2C_CTL0 must be cleared. Before entering Deep-sleep mode (I2CEN=0), the I2C peripheral must be disabled if wakeup from Deep-sleep mode is disabled (WUEN = 0).

Note: Only address match of I2C0 can wakeup MCU from Deep-sleep mode.

19.3.12. 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.

19.3.13. I2C error and interrupts

The I2C error flags are listed in [Table 19-6. I2C error flags](#).

Table 19-6. 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 19-7. I2C interrupt events](#).

Table 19-7. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP condition 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	

19.3.14. 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 I2Cx_HOLD configuration bits in the DBG module.

19.4. Register definition

I2C0 secure access base address: 0x5000 5400

I2C0 non-secure access base address: 0x4000 5400

I2C1 secure access base address: 0x5000 5800

I2C1 non-secure access base address: 0x4000 5800

19.4.1. Control register 0 (I2C_CTL0)

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
								PECEN	SMBALT	SMBDAE	SMBHAE	GCEN	WUEN	SS	SBCTL
								EN	N	N					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	Reserved	ANOFF		DNF[3:0]		ERRIE	TCIE	STPDETE	NACKIE	ADDMIE	RBNEIE	TIE		I2CEN
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	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	SMBHAE	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	<p>Whether or not to response to a General Call (0x00)</p> <p>0: Slave won't response to a General Call</p> <p>1: Slave will response to a General Call</p>
18	WUEN	<p>Wakeup from Deep-sleep mode enable</p> <p>0: Wakeup from Deep-sleep mode disable.</p> <p>1: Wakeup from Deep-sleep mode enable.</p> <p>Note: WUEN can be set only when DNF[3:0] = 0000. This bit is reserved in I2C1.</p>
17	SS	<p>Whether to stretch SCL low when data is not ready in slave mode.</p> <p>This bit is set and cleared by software.</p> <p>0: SCL Stretching is enabled</p> <p>1: SCL Stretching is disabled</p> <p>Note: When in master mode, this bit must be 0. This bit can be modified when I2CEN = 0.</p>
16	SBCTL	<p>Slave byte control</p> <p>This bit is used to enable hardware byte control in slave mode.</p> <p>0: Slave byte control is disabled</p> <p>1: Slave byte control is enabled</p>
15	DENR	<p>DMA enable for reception</p> <p>0: DMA is disabled for reception</p> <p>1: DMA is enabled for reception</p>
14	DENT	<p>DMA enable for transmission</p> <p>0: DMA is disabled for transmission</p> <p>1: DMA is enabled for transmission</p>
13	Reserved	Must be kept at reset value.
12	ANOFF	<p>Analog noise filter disable</p> <p>0: Analog noise filter is enabled</p> <p>1: Analog noise filter is disabled</p> <p>Note: This bit can only be programmed when the I2C is disabled (I2CEN = 0).</p>
11:8	DNF[3:0]	<p>Digital noise filter</p> <p>These bits are used to configure the digital noise filter on SDA and SCL input. The digital filter will filter spikes with a length of up to $DNF[3:0] * t_{I2CCLK}$</p> <p>0000: Digital filter is disabled</p> <p>0001: Digital filter is enabled and filter spikes with a length of up to 1 t_{I2CCLK}</p> <p>...</p> <p>1111: Digital filter is enabled and filter spikes with a length of up to 15 t_{I2CCLK}</p> <p>These bits can only be modified when the I2C is disabled (I2CEN = 0).</p>
7	ERRIE	<p>Error interrupt enable</p> <p>0: Error interrupt disabled</p>

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 acknol edge received interrupt enable 0: Not acknol edge (NACK) received interrupt is disabled 1: Not acknol edge (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

19.4.2. Control register 1 (I2C_CTL1)

Address offset: 0x04

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								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	<p>PEC Transfer</p> <p>Set by software.</p> <p>Cleared by hardware in the following cases:</p> <p>When PEC byte is transferred or ADDSEND bit is set or STOP condition is detected or I2CEN=0.</p> <p>0: Don't transfer PEC value</p> <p>1: Transfer PEC</p> <p>Note: This bit has no effect when RELOAD=1, or SBCTL=0 in slave mode.</p>
25	AUTOEND	<p>Automatic end mode in master mode</p> <p>0: TC bit is set when the transfer of BYTENUM[7:0] bytes is completed.</p> <p>1: a STOP condition is sent automatically when the transfer of BYTENUM[7:0] bytes is completed.</p> <p>Note: This bit works only when RELOAD=0. This bit is set and cleared by software.</p>
24	RELOAD	<p>Reload mode</p> <p>0: After the data of BYTENUM[7:0] bytes transfer, the transfer is completed.</p> <p>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.</p> <p>This bit is set and cleared by software.</p>
23:16	BYTENUM[7:0]	<p>Number of bytes to be transferred</p> <p>These bits are programmed with the number of bytes to be transferred. When SBCTL=0, these bits have no effect.</p> <p>Note: These bits should not be modified when the START bit is set.</p>
15	NACKEN	<p>Generate NACK in slave mode</p> <p>0: an ACK is sent after receiving a new byte.</p> <p>1: a NACK is sent after receiving a new byte.</p> <p>Note: The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP condition 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.</p>
14	STOP	<p>Generate a STOP condition on I2C bus</p> <p>This bit is set by software and cleared by hardware when I2CEN=0 or STOP condition is detected.</p> <p>0: STOP will not be sent</p> <p>1: STOP will be sent</p>
13	START	<p>Generate a START condition on I2C bus</p> <p>This bit is set by software and cleared by hardware after the address is sent. When</p>

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 ADDSENDC 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, this bit can not be changed.
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, this bit can not be modified.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive Note: When the START bit is set, this bit can not be modified.
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 have 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, the bit filed can not be modified.

19.4.3. Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

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
ADDRESSEN SEN	Reserved			ADDFORMAT	ADDRESS[9:8]					ADDRESS[7:1]				ADDRESS0 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: When ADDRESSEN is set, this bit should not be written.
0	ADDRESS0	Bit 0 of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.

19.4.4. Slave address register 1 (I2C_SADDR1)

Address offset: 0x0C

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
ADDRESSEN S2EN	Reserved				ADDMASK2[2:0]			ADDRESS2[7:1]					Reserved		
rw					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:0] 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: When ADDRESS2EN is set, these bits should not be written.
0	Reserved	Must be kept at reset value.

19.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

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
PSC[3:0]				Reserved				SCLDELY[3:0]				SDADELY[3:0]			
rw								rw					rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

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. $t_{PSC} = (PSC+1) \times t_{I2CCLK}$
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.

$$t_{SCLDELY} = (SCLDELY + 1) \times t_{PSC}$$

19:16	SDADELAY[3:0]	Data hold time
		A delay $t_{SDADELAY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELAY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0.
		$t_{SDADELAY} = SDADELAY \times t_{PSC}$
15:8	SCLH[7:0]	SCL high period
		SCL high period can be generated by configuring these bits.
		$t_{SCLH} = (SCLH+1) \times t_{PSC}$
		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.
		$t_{SCLL} = (SCLL+1) \times t_{PSC}$
		Note: These bits can only be used in master mode.

19.4.6. Timeout register (I2C_TIMEOUT)

Address offset: 0x14

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
EXTOEN	Reserved		BUSTOB[11:0]												
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	Reserved		TOIDLE		BUSTOA[11:0]										
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	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.

$$t_{LOW:EXT} = (BUSTOB +1) \times 2048 \times t_{I2CCLK}$$

Note: These bits can be modified only when EXTOEN =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	Bus timeout A When $TOIDLE=0$, $t_{TIMEOUT}=(BUSTOA +1) \times 2048 \times t_{I2CCLK}$ When $TOIDLE=1$, $t_{IDLE} = (BUSTOA +1) \times 4 \times t_{I2CCLK}$ Note: These bits can be written only when TOEN =0.

19.4.7. Status register (I2C_STAT)

Address offset: 0x18

Reset value: 0x0000 0001

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										READDR[6:0]					TR
r										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	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

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 condition is detected and cleared by hardware after a STOP condition. 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, SMBALTN=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	Overrun/Underrun error in slave mode In slave mode with SS=1, when an overrun/underrun 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 overrun or underrun occurs 1: Overrun or underrun 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 condition 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 condition detected in slave mode This flag is set by hardware when a STOP condition is detected on the bus. It is cleared by software by setting STPDETC bit and cleared by hardware when I2CEN=0. 0: STOP condition is not detected. 1: STOP condition 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 ADDSENDC 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

19.4.8. Status clear register (I2C_STATC)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SMBALT C	TIMEOUT C	PECERR C	OUERRC	LOSTAR BC	BERRC	Reserved	STPDET C	NACKC	ADDSEN DC					Reserved
	w	w	w	w	w	w		w	w	w					

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

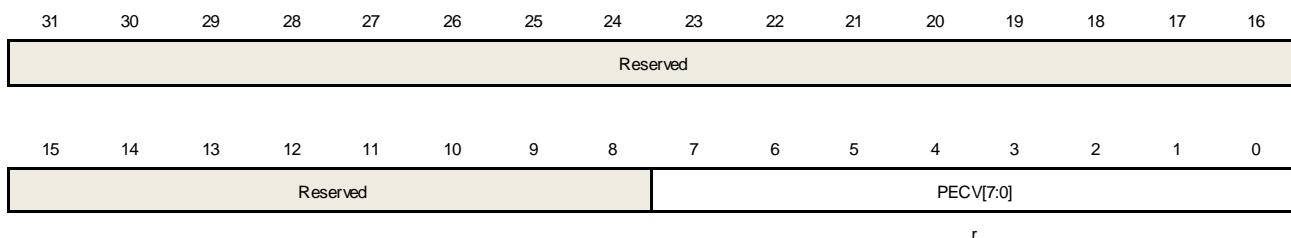
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit
7:6	Reserved	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by write 1 to this bit
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by write 1 to this bit
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by write 1 to this bit
2:0	Reserved	Must be kept at reset value.

19.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

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



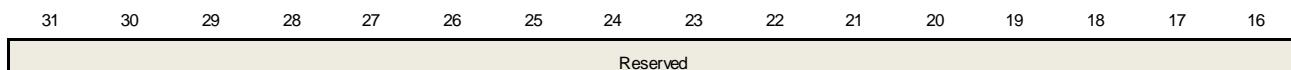
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

19.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

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



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

19.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

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								TDATA [7:0]							

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

19.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

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
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.

20. Serial peripheral interface/Inter-IC sound (SPI/I2S)

20.1. Overview

The SPI/I2S module can communicate with external devices using the SPI protocol or the I2S audio 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 only supported in SPI0.

The inter-IC sound (I2S) supports four audio standards: I2S Phillips standard, MSB justified standard, LSB justified standard, and PCM standard. I2S works at either master or slave mode for transmission and reception. (By using I2S1_ADD module, full duplex mode communication is realized in I2S1.)

20.2. Characteristics

20.2.1. SPI characteristics

- Master or slave operation with full-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.
- Quad-SPI configuration available in master mode (only in SPI0).

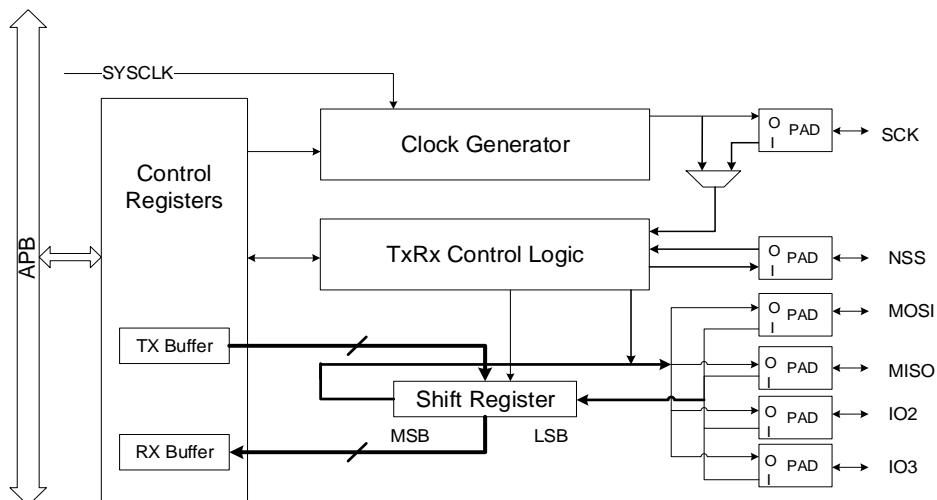
20.2.2. I2S characteristics

- Master or slave operation for transmission/reception.
- Master or slave operation with full-duplex mode(only in I2S1)
- Four I2S standards supported: Phillips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 bits or 32 bits.
- Transmission and reception using a 16 bits wide buffer.
- Audio sample frequency can be 8 kHz to 192 kHz using I2S clock divider.
- Programmable idle state clock polarity.
- Master clock (MCK) can be output.

- Transmission and reception using DMA.

20.3. SPI block diagram

Figure 20-1. Block diagram of SPI



20.4. SPI signal description

20.4.1. Normal configuration (Not Quad-SPI Mode)

Table 20-1. 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: NSS output for single master (NSSDRV=1) or for multi-master (NSSDRV=0) application.

Pin name	Direction	Description
		Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

20.4.2. 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 (only available in SPI0). Quad-SPI mode can only work in master mode.

The IO2 and IO3 pins can be driven high in normal Non-Quad-SPI mode by configuring IO23_DRV bit in SPI_QCTL register.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 20-2. Quad-SPI signal description

Pin name	Direction	Description
SCK	O	SPI clock output
MOSI	IO	Transmission/Reception data 0
MISO	IO	Transmission/Reception data 1
IO2	IO	Transmission/Reception data 2
IO3	IO	Transmission/Reception data 3
NSS	O	NSS output

20.5. SPI function overview

20.5.1. 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 idle and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

Figure 20-2. SPI timing diagram in normal mode

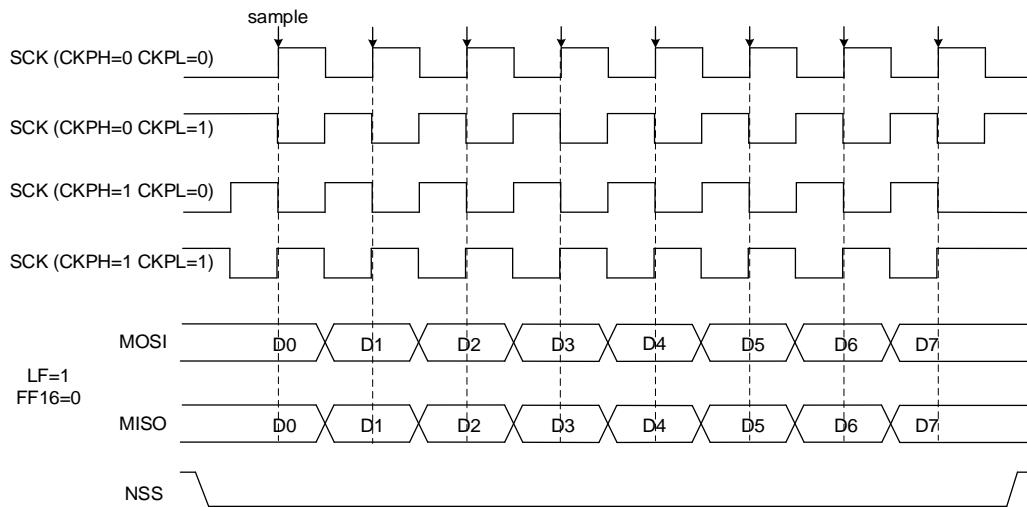
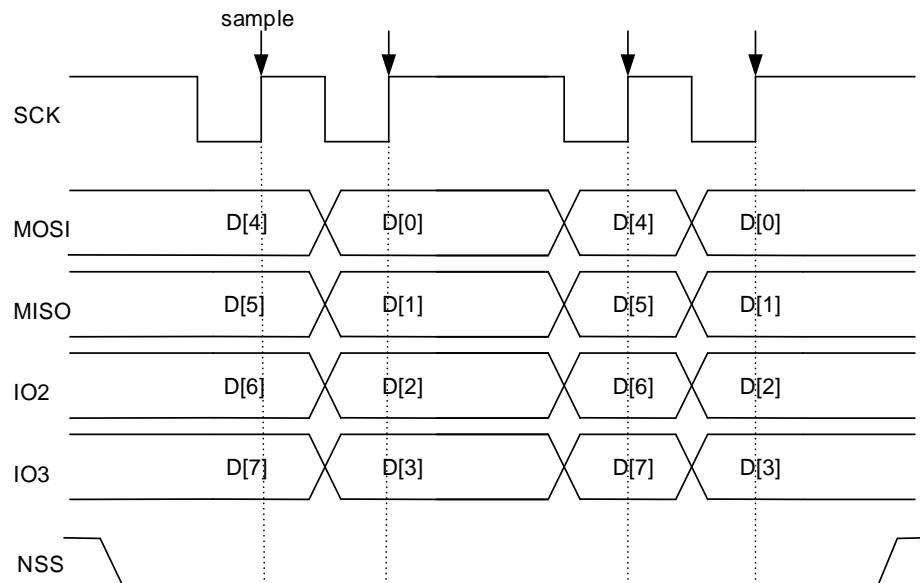


Figure 20-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



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.

20.5.2. NSS function

Slave mode

When slave mode is configured (MSTMOD=0), SPI gets NSS level from NSS pin in hardware

NSS mode (SWNSSEN=0) or from SWNSS bit in software NSS mode (SWNSSEN=1), and SPI transmits/receives data only when NSS level is low. In software NSS mode, NSS pin is not used.

Master mode

In master mode (MSTMOD=1), if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSEN=0, NSSDRV=0) or software mode (SWNSSEN=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 stays high after SPI is enabled and goes low when transmission or reception process begins. When SPI is disabled, the NSS goes high.

The application may also use a general purpose IO as NSS pin to realize more flexible NSS.

20.5.3. SPI operating modes

Table 20-3. 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 unidirectional connection	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Not used 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

Mode	Description	Register configuration	Data pin usage
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 20-4. A typical full-duplex connection

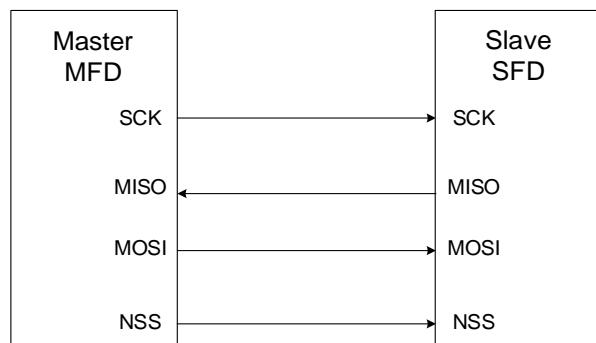


Figure 20-5. A typical simplex connection (Master: Receive, Slave: Transmit)

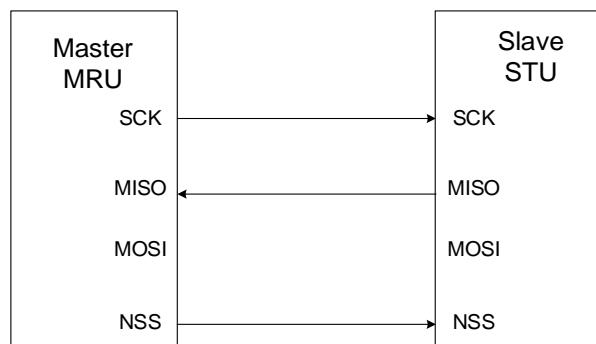


Figure 20-6. A typical simplex connection (Master: Transmit only, Slave: Receive)

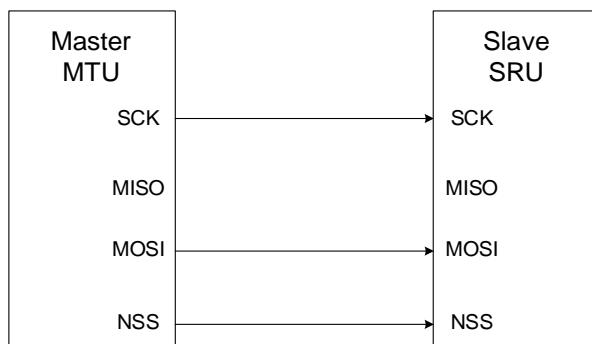
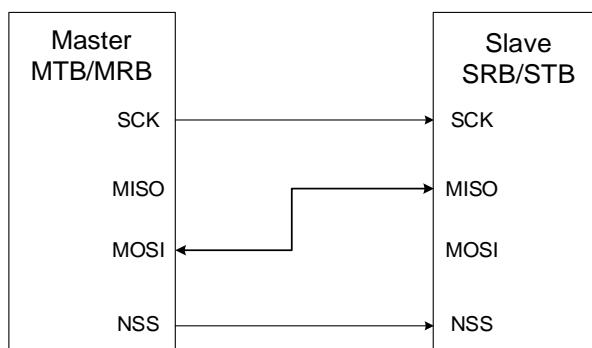


Figure 20-7. A typical bidirectional connection



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. Configure MSTMOD, RO, BDEN and BDOEN depending on the operating modes described in [SPI operating modes](#) section.
8. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
9. Enable the SPI (set the SPIEN bit).

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 regardless of the RBNE and OVRE bits.

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 regardless of the TBE flag.

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 20-8. Timing diagram of TI master mode with discontinuous transfer

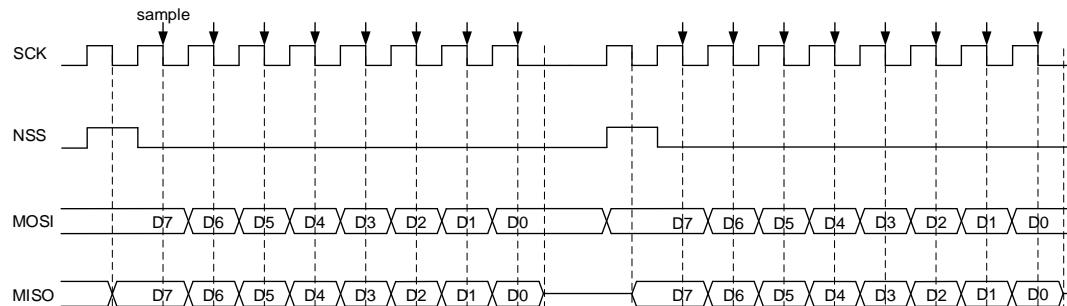
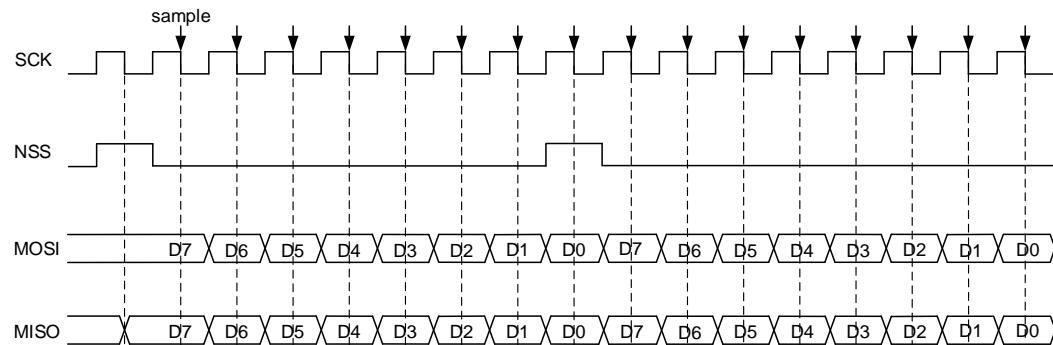
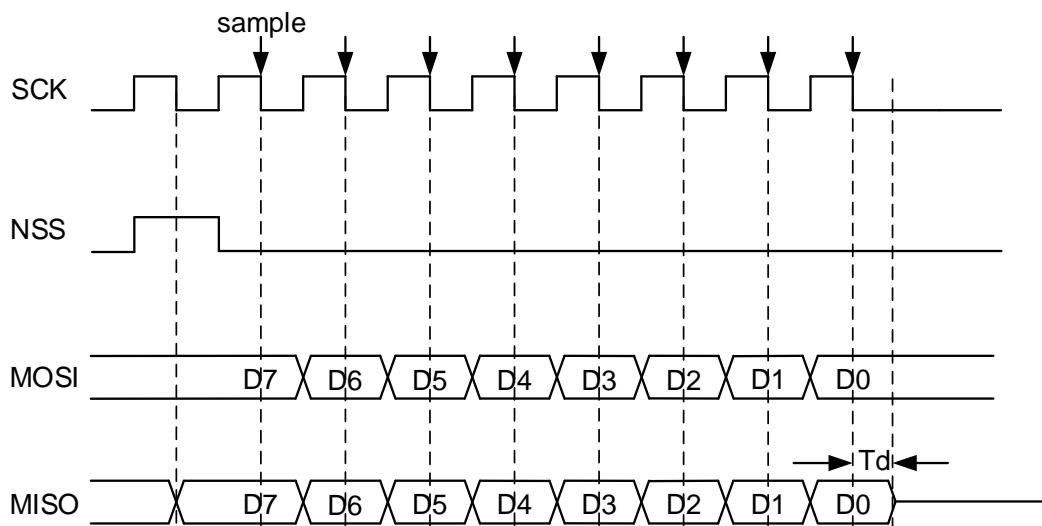


Figure 20-9. Timing diagram of TI master mode with continuous transfer



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 previous bytes.

Figure 20-10. 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.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (20-1)$$

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.

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

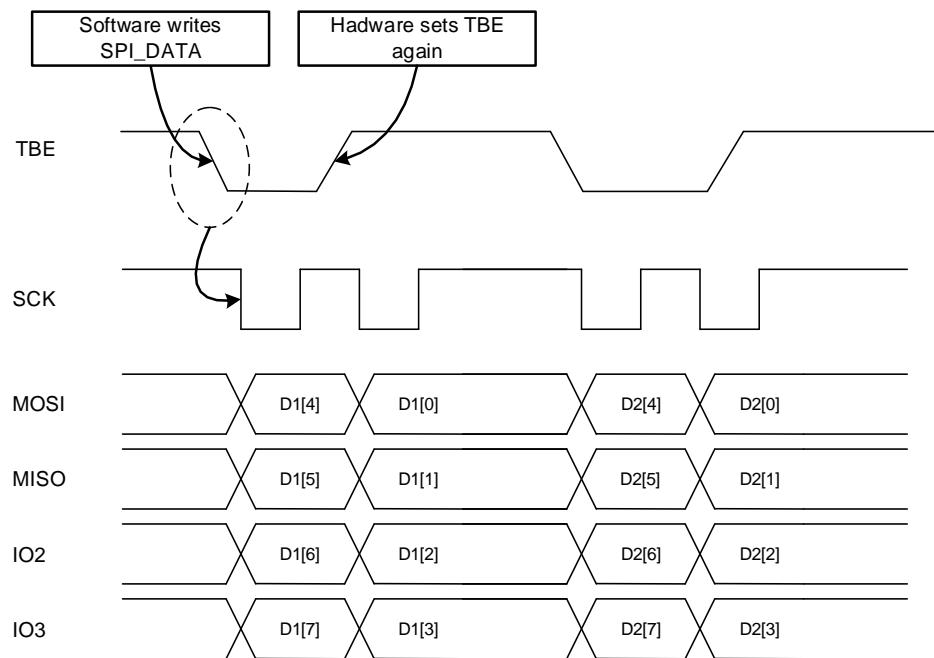
SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and 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 of data 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 20-11. Timing diagram of quad write operation in Quad-SPI mode



Quad read operation

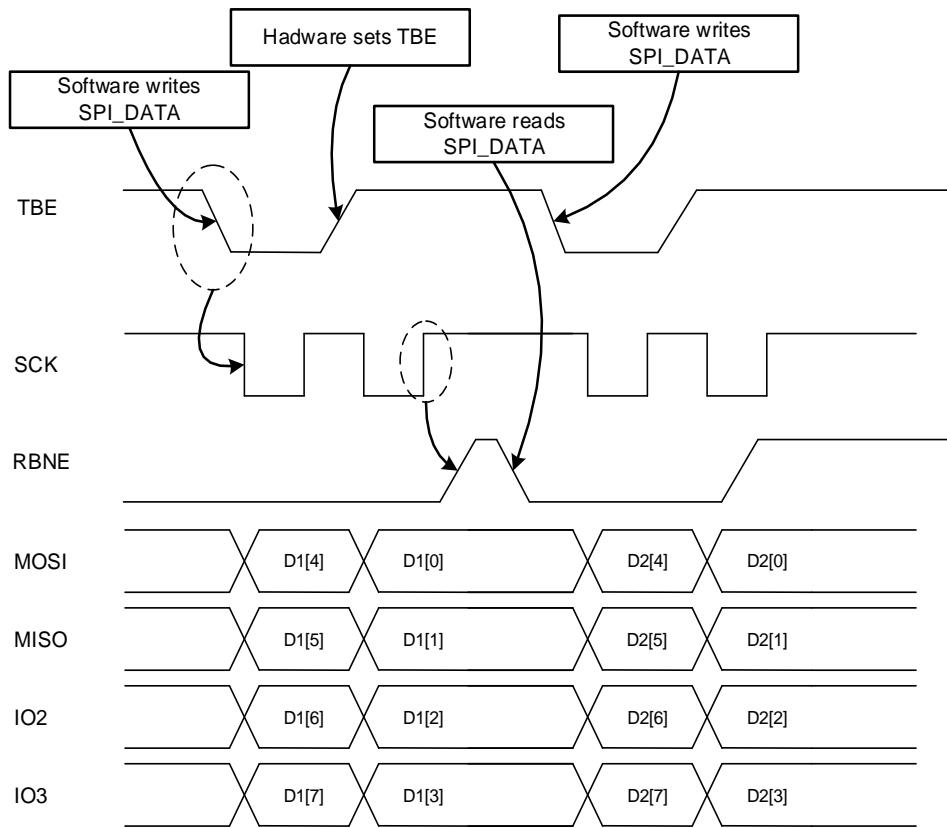
SPI works in quad read mode when QMOD and QRD bits are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Writing data into SPI_DATA is 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 20-12. 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 completes.

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.

20.5.4. 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 correctly configure DMA modules, 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.

20.5.5. 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 continuously calculate CRC for each bit transmitted and received on lines, 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.

20.6. SPI interrupts

20.6.1. 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.

20.6.2. 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)

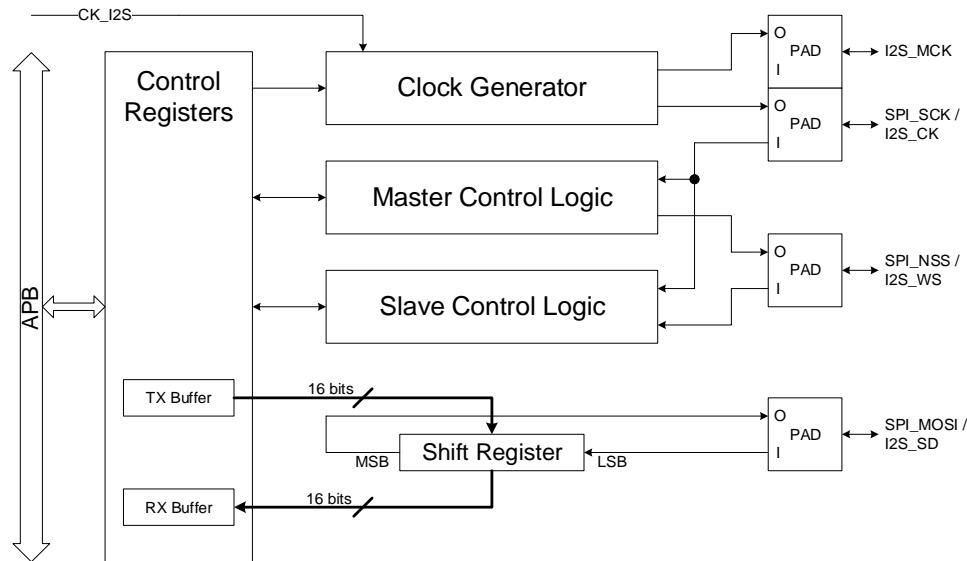
When the CRCEN bit is set, the CRC calculation result of the received data in the SPI_RCRC register is compared with the received CRC value after the last data, the CRCERR is set when they are different.

Table 20-4. 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	

20.7. I2S block diagram

Figure 20-13. Block diagram of I2S



There are five sub modules to support I2S function, including control registers, clock generator, master control logic, slave control logic and shift register. All the user configuration registers are implemented in the control registers module, including the TX buffer and RX buffer. The clock generator is used to produce I2S communication clock in master mode. The master control logic is implemented to generate the I2S_WS signal and control the communication in master mode. The slave control logic is implemented to control the communication in slave mode according to the received I2S_CK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

20.8. I2S signal description

There are four pins on the I2S interface, including I2S_CK, I2S_WS, I2S_SD and I2S_MCK. I2S_CK is the serial clock signal, which shares the same pin with SPI_SCK. I2S_WS is the frame control signal, which shares the same pin with SPI_NSS. I2S_SD is the serial data signal, which shares the same pin with SPI_MOSI. I2S_MCK is the master clock signal. It produces a frequency rate equals to $256 \times F_s$, and F_s is the audio sampling frequency.

20.9. I2S function overview

20.9.1. I2S audio standards

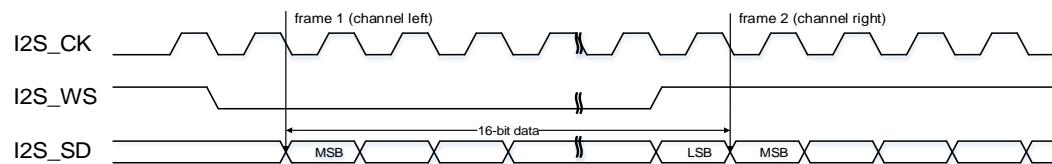
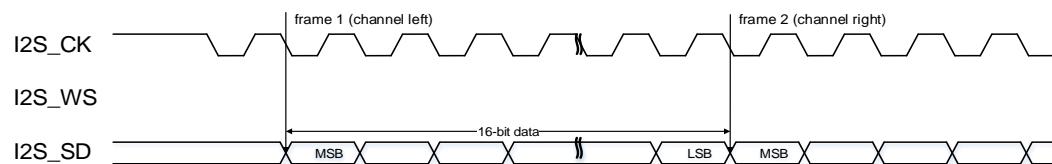
The I2S audio standard is selected by the I2SSTD bits in the SPI_I2SCTL register. Four audio standards are supported, including I2S Phillips standard, MSB justified standard, LSB justified standard, and PCM standard. All standards except PCM handle audio data time-multiplexed on two channels (the left channel and the right channel). For these standards, the I2S_WS signal indicates the channel side. For PCM standard, the I2S_WS signal indicates frame synchronization information.

The data length and the channel length are configured by the DTLEN bits and CHLEN bit in the SPI_I2SCTL register. Since the channel length must be greater than or equal to the data length, four packet types are available. They are 16-bit data packed in 16-bit frame, 16-bit data packed in 32-bit frame, 24-bit data packed in 32-bit frame, and 32-bit data packed in 32-bit frame. The data buffer for transmission and reception is 16-bit wide. In the case that the data length is 24 bits or 32 bits, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In the case that the data length is 16 bits, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. When using 16-bit data packed in 32-bit frame, 16-bit 0 is inserted by hardware automatically to extend the data to 32-bit format.

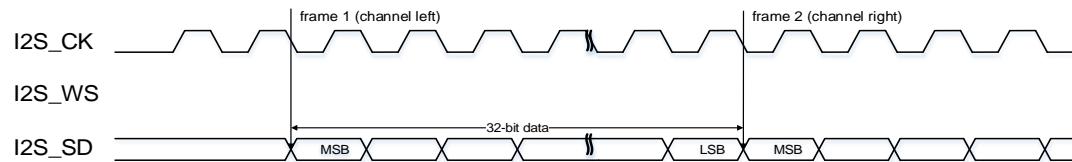
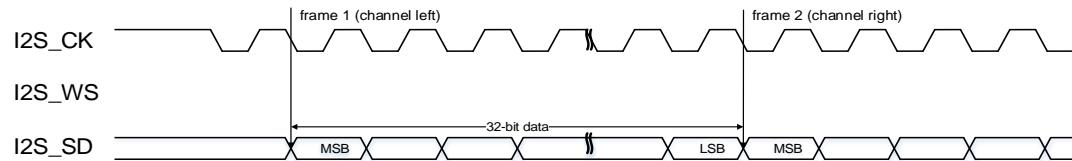
For all standards and packet types, the most significant bit (MSB) is always sent first. For all standards based on two channels time-multiplexed, the channel left is always sent first followed by the channel right.

I2S Phillips standard

For I2S Phillips standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The timing diagrams for each configuration are shown below.

Figure 20-14. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

Figure 20-15. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)


When the packet type is 16-bit data packed in 16-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame.

Figure 20-16. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

Figure 20-17. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)


When the packet type is 32-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 32-bit data is going to be sent, the first data written to the SPI_DATA register should be the higher 16 bits, and the second one should be the lower 16 bits. In reception mode, if a 32-bit data is received, the first data read from the SPI_DATA register should be the higher 16 bits, and the second one should be the lower 16 bits.

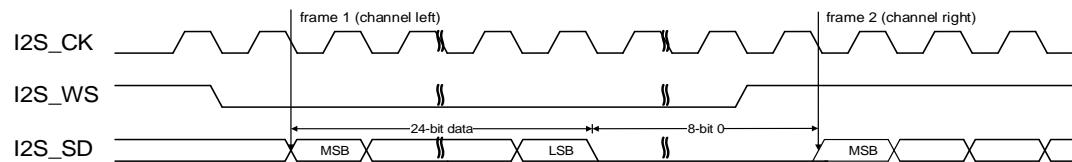
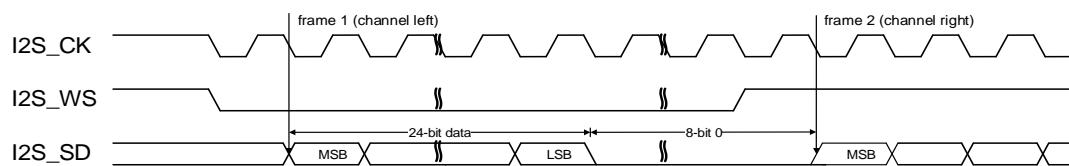
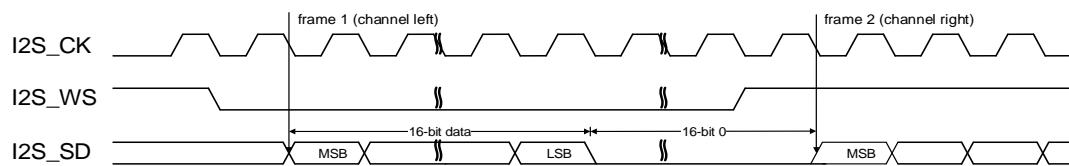
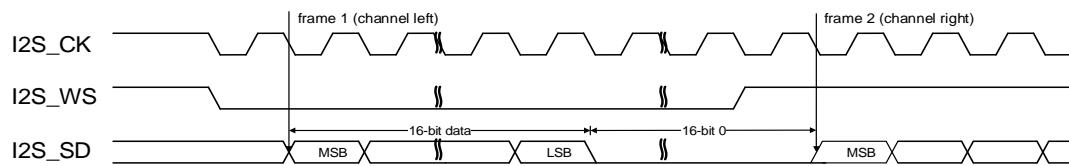
Figure 20-18. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)


Figure 20-19. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)


When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 24-bit data D[23:0] is going to be sent, the first data written to the SPI_DATA register should be the higher 16 bits D[23:8]. And the second one should be a 16-bit data, the higher 8 bits of this 16-bit data should be D[7:0] and the lower 8 bits can be any value. In reception mode, if a 24-bit data D[23:0] is received, the first data read from the SPI_DATA register is D[23:8]. And the second one is a 16-bit data, the higher 8 bits of this 16-bit data are D[7:0] and the lower 8 bits are zeros.

Figure 20-20. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

Figure 20-21. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)


When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

MSB justified standard

For MSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The SPI_DATA register is handled in the exactly same way as that for I2S Phillips standard. The timing diagrams for each configuration are shown below.

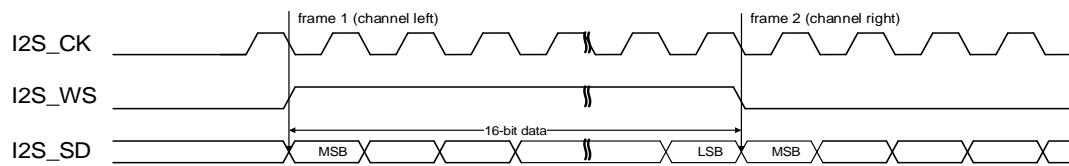
Figure 20-22. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)


Figure 20-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

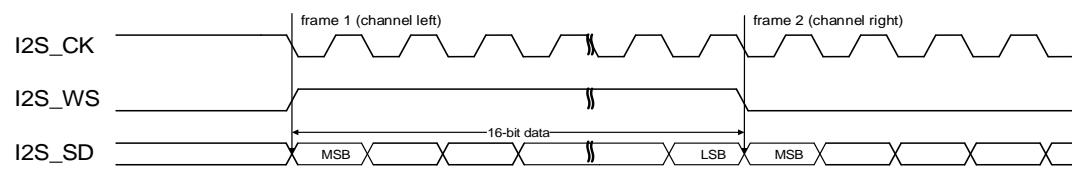


Figure 20-24. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

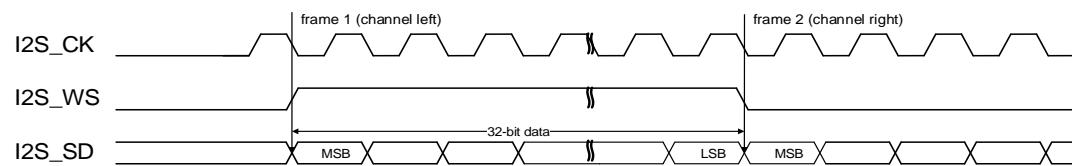


Figure 20-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

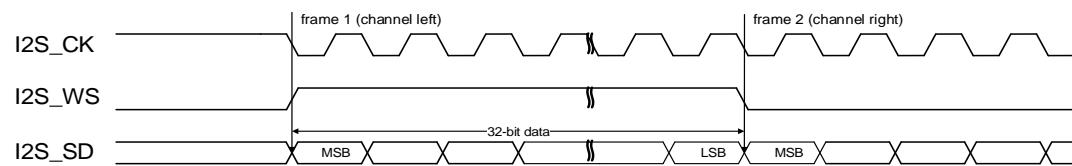


Figure 20-26. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

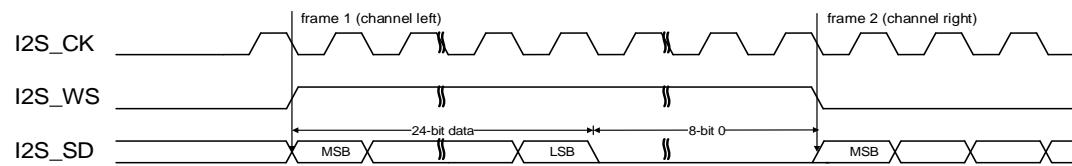


Figure 20-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

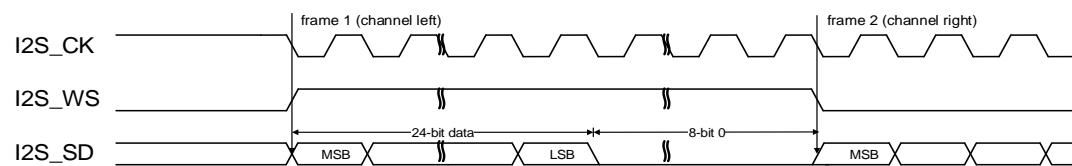


Figure 20-28. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

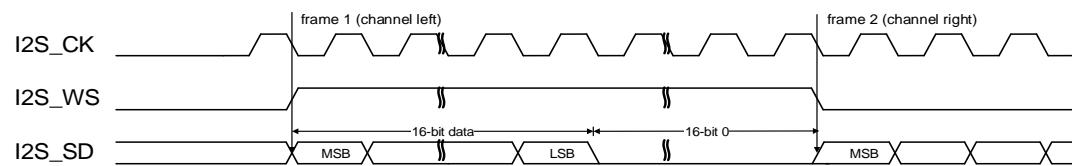
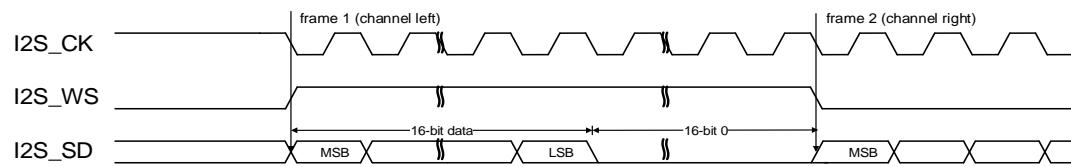
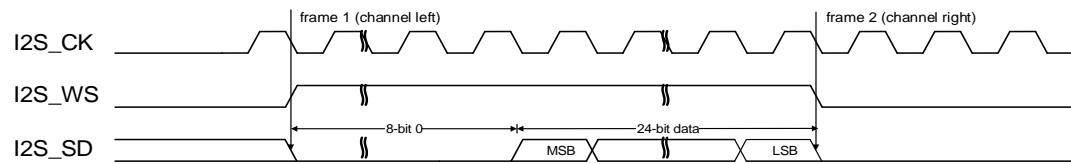
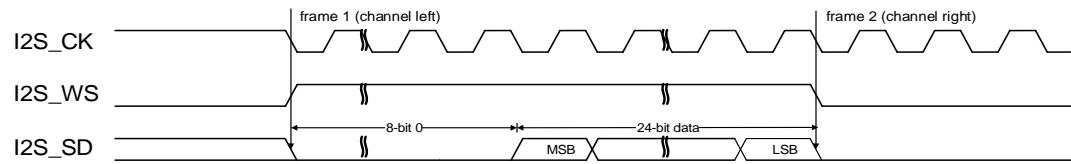


Figure 20-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)


LSB justified standard

For LSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. In the case that the channel length is equal to the data length, LSB justified standard and MSB justified standard are exactly the same. In the case that the channel length is greater than the data length, the valid data is aligned to LSB for LSB justified standard while the valid data is aligned to MSB for MSB justified standard. The timing diagrams for the cases that the channel length is greater than the data length are shown below.

Figure 20-30. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

Figure 20-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)


When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 24-bit data D[23:0] is going to be sent, the first data written to the SPI_DATA register should be a 16-bit data. The higher 8 bits of the 16-bit data can be any value and the lower 8 bits should be D[23:16]. The second data written to the SPI_DATA register should be D[15:0]. In reception mode, if a 24-bit data D[23:0] is received, the first data read from the SPI_DATA register is a 16-bit data. The high 8 bits of this 16-bit data are zeros and the lower 8 bits are D[23:16]. The second data read from the SPI_DATA register is D[15:0].

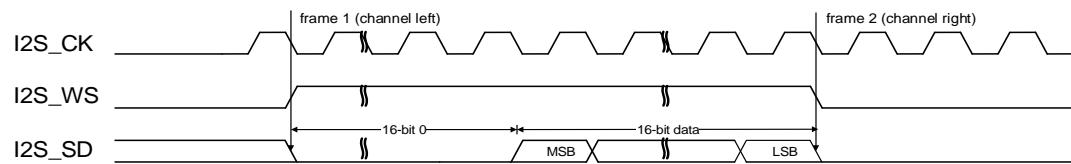
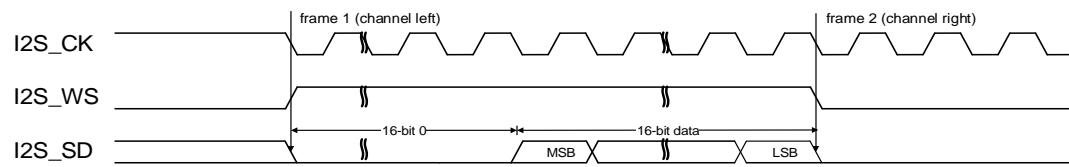
Figure 20-32. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)


Figure 20-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

PCM standard

For PCM standard, I2S_WS and I2S_SD are updated on the rising edge of I2S_CK, and the I2S_WS signal indicates frame synchronization information. Both the short frame synchronization mode and the long frame synchronization mode are available and configurable using the PCMSMOD bit in the SPI_I2SCTL register. The SPI_DATA register is handled in the exactly same way as that for I2S Phillips standard. The timing diagrams for each configuration of the short frame synchronization mode are shown below.

Figure 20-34. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

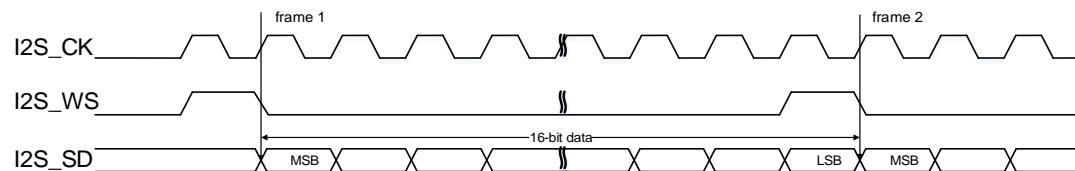


Figure 20-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

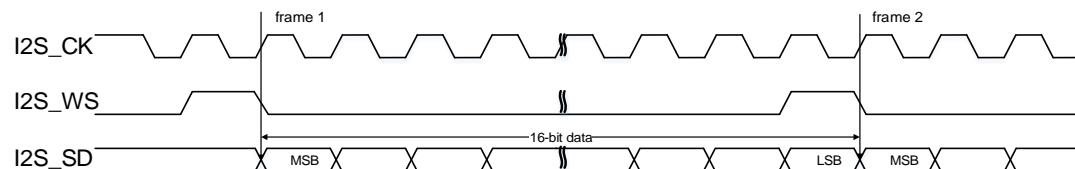


Figure 20-36. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

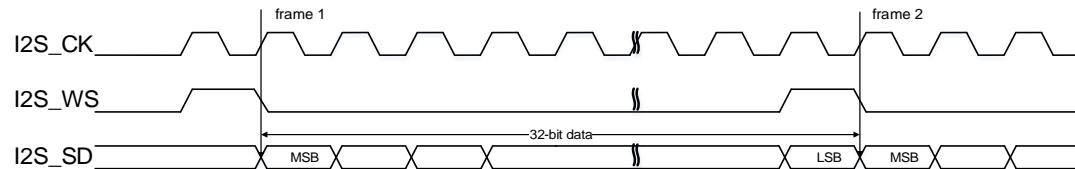


Figure 20-37. PCM standard short frame synchronization mode timing diagram

(DTLEN=10, CHLEN=1, CKPL=1)

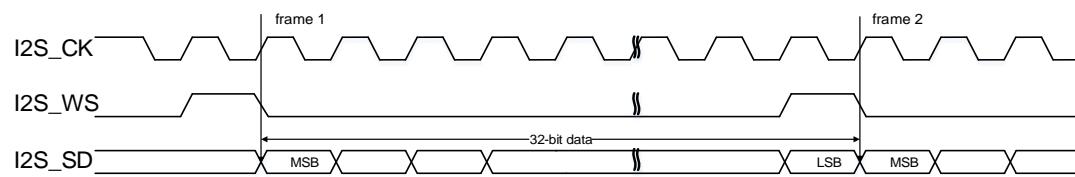


Figure 20-38. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

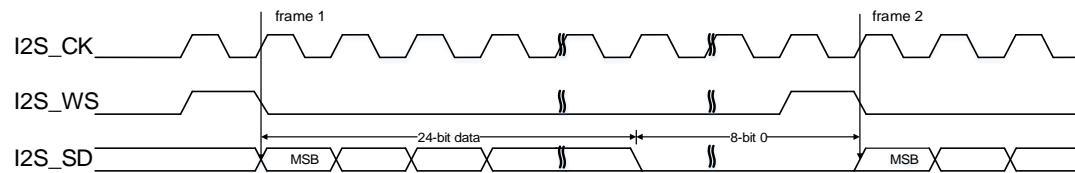


Figure 20-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

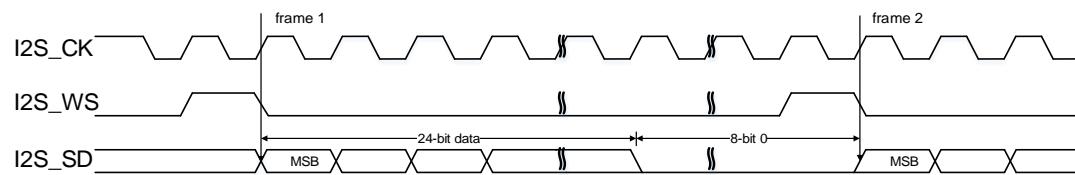


Figure 20-40. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

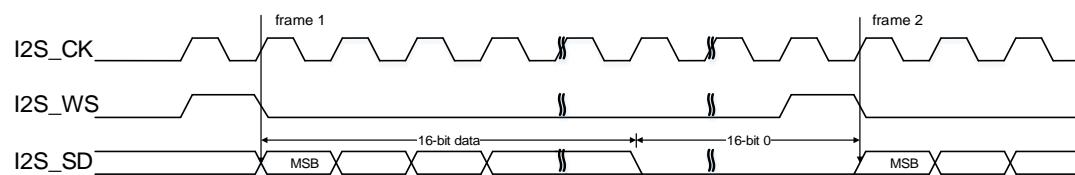
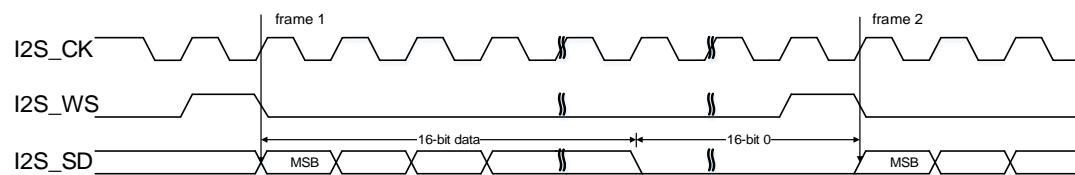


Figure 20-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 20-42. PCM standard long frame synchronization mode timing diagram

(DTLEN=00, CHLEN=0, CKPL=0)

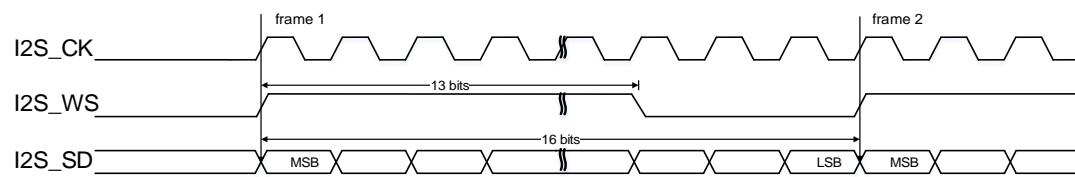


Figure 20-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

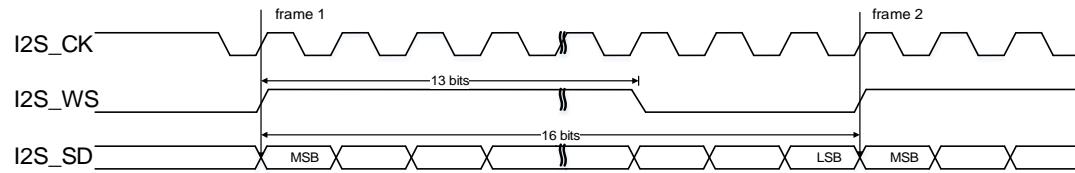


Figure 20-44. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=0, CKPL=0)

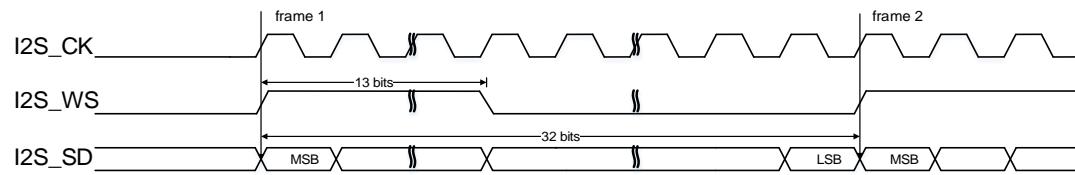


Figure 20-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

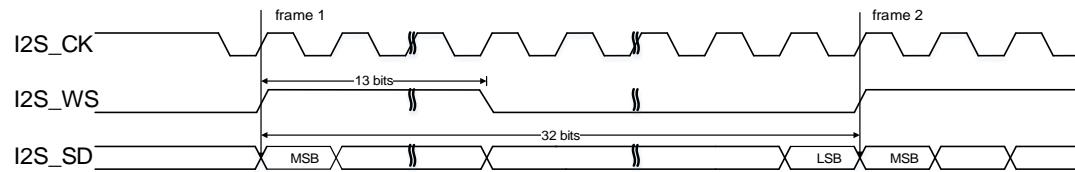


Figure 20-46. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

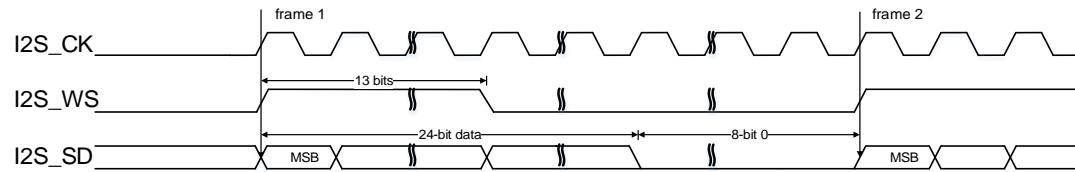


Figure 20-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

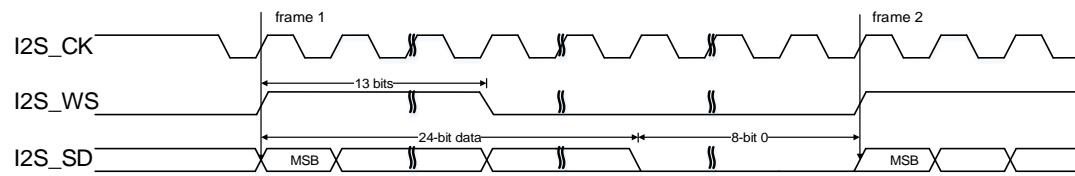


Figure 20-48. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

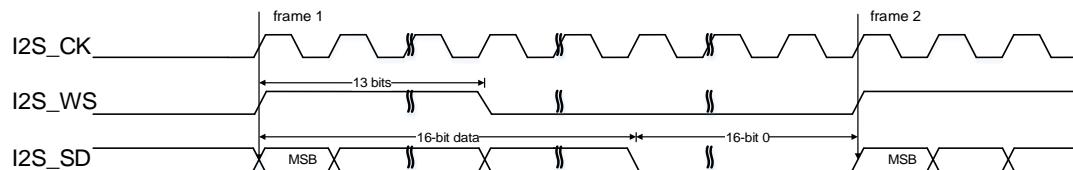
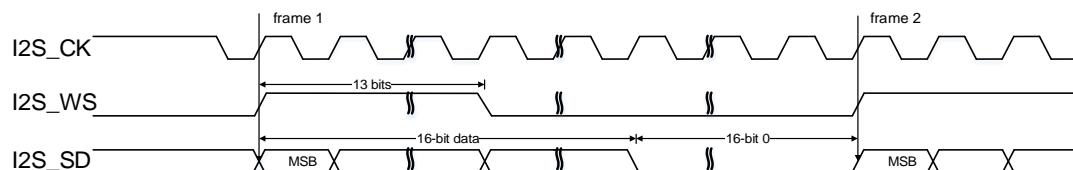
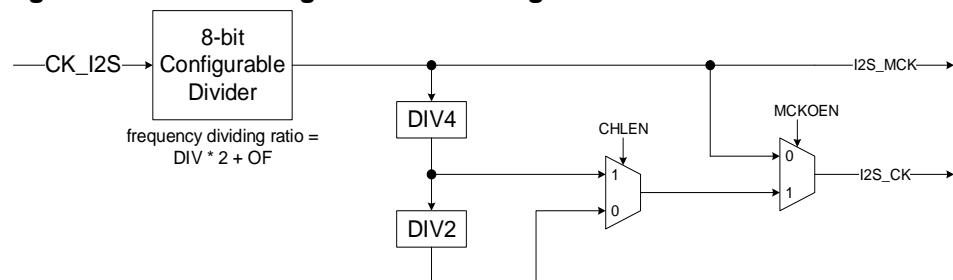


Figure 20-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



20.9.2. I2S clock

Figure 20-50. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as [Figure 20-50. Block diagram of I2S clock generator](#). The I2S interface clocks are configured by the DIV bits, the OF bit, the MCKOEN bit in the SPI_I2SPSC register and the CHLEN bit in the SPI_I2SCTL register. The I2S bitrate can be calculated by the formulas shown in [Table 20-5. I2S bitrate calculation formulas](#).

Table 20-5. I2S bitrate calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

The relationship between audio sampling frequency (F_s) and I2S bitrate is defined by the following formula:

$$F_s = \text{I2S bitrate} / (\text{number of bits per channel} * \text{number of channels})$$

So, in order to get the desired audio sampling frequency, the clock generator needs to be configured according to the formulas listed in [Table 20-6. Audio sampling frequency calculation formulas](#).

Table 20-6. Audio sampling frequency calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

20.9.3. Operation

Operation modes

The operation mode is selected by the I2SOPMOD bits in the SPI_I2SCTL register. There are four available operation modes, including master transmission mode, master reception mode, slave transmission mode, and slave reception mode. The direction of I2S interface signals for each operation mode is shown in the [Table 20-7. Direction of I2S interface signals for each operation mode](#).

Table 20-7. Direction of I2S interface signals for each operation mode

Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD	I2S_ADD_SD(2)
Master transmission	output or NU(1)	output	output	output	NU(1)
Master reception	output or NU(1)	output	output	input	NU(1)
Slave transmission	input or NU(1)	input	input	output	NU(1)
Slave reception	input or NU(1)	input	input	input	NU(1)
Full-duplex	output or NU(1)	output	output	output or input	Input or output

1. NU means the pin is not used by I2S and can be used by other functions.
2. In order to support the full-duplex operation mode, I2S1 requires an additional on-chip I2S module: I2S_ADD1. The I2S_ADD_SD pin is the data pin of the I2S_ADD module. The full-duplex mode will be described in detail in this chapters.

I2S initialization sequence

I2S initialization sequence contains five steps shown below. In order to initialize I2S to master mode, all the five steps should be done. In order to initialize I2S to slave mode, only step 2,

step 3, step 4 and step 5 should be done.

- Step 1: Configure the DIV[7:0] bits, the OF bit, and the MCKOEN bit in the SPI_I2SPSC register to define the I2S bitrate and determine whether I2S_MCK needs to be provided or not.
- Step 2: Configure the CKPL in the SPI_I2SCTL register to define the idle state clock polarity.
- Step 3: Configure the I2SSEL bit, the I2SSTD[1:0] bits, the PCMSMOD bit, the I2SOPMOD[1:0] bits, the DTLEN[1:0] bits, and the CHLEN bit in the SPI_I2SCTL register to define the I2S feature.
- Step 4: Configure the TBEIE bit, the RBNEIE bit, the ERRIE bit, the DMATEN bit, and the DMAREN bit in the SPI_CTL1 register to select the potential interrupt sources and the DMA capabilities. This step is optional.
- Step 5: Set the I2SEN bit in the SPI_I2SCTL register to enable I2S.

I2S master transmission sequence

The TBE flag is used to control the transmission sequence. As is mentioned before, the TBE flag indicates that the transmit buffer is empty, and an interrupt will be generated if the TBEIE bit in the SPI_CTL1 register is set. At the beginning, the transmit buffer is empty (TBE is high) and no transmission sequence is processing in the shift register. When a half word is written to the SPI_DATA register (TBE goes low), the data is transferred from the transmit buffer to the shift register (TBE goes high) immediately. At the moment, the transmission sequence begins.

The data is parallel loaded into the 16-bit shift register, and shifted out serially to the I2S_SD pin, MSB first. The next data should be written to the SPI_DATA register, when the TBE flag is high. After a write operation to the SPI_DATA register, the TBE flag goes low. When the current transmission finishes, the data in the transmit buffer is loaded into the shift register, and the TBE flag goes back high. Software should write the next audio data into SPI_DATA register before the current data finishes, otherwise, the audio data transmission is not continuous.

For all standards except PCM, the I2SCH flag is used to distinguish which channel side the data to transfer belongs to. The I2SCH flag is refreshed at the moment when the TBE flag goes high. At the beginning, the I2SCH flag is low, indicating the left channel data should be written to the SPI_DATA register.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

I2S master reception sequence

The RBNE flag is used to control the reception sequence. As is mentioned before, the RBNE flag indicates the receive buffer is not empty, and an interrupt will be generated if the RBNEIE bit in the SPI_CTL1 register is set. The reception sequence begins immediately when the I2SEN bit in the SPI_I2SCTL register is set. At the beginning, the receive buffer is empty

(RBNE is low). When a reception sequence finishes, the received data in the shift register is loaded into the receive buffer (RBNE goes high). The data should be read from the SPI_DATA register, when the RBNE flag is high. After a read operation to the SPI_DATA register, the RBNE flag goes low. It is mandatory to read the SPI_DATA register before the end of the next reception. Otherwise, reception overrun error occurs. The RXORERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is necessary to disable and then enable I2S before resuming the communication.

For all standards except PCM, the I2SCH flag is used to distinguish which channel side the received data belongs to. The I2SCH flag is refreshed at the moment when the RBNE flag goes high.

Different sequences are used to disable the I2S in different standards, data length and channel length. The sequences for each case are described below.

- 16-bit data packed in 32-bit frame in the LSB justified standard (DTLEN = 00, CHLEN = 1, and I2SSTD = 10)
 1. Wait for the second last RBNE.
 2. Then wait 17 I2S CK clock (clock on I2S_CK pin) cycles.
 3. Clear the I2SEN bit.
- 16-bit data packed in 32-bit frame in the audio standards except the LSB justified standard (DTLEN = 00, CHLEN = 1, and I2SSTD is not equal to 0b10)
 1. Wait for the last RBNE.
 2. Then wait one I2S clock cycle.
 3. Clear the I2SEN bit.
- For all other cases
 1. Wait for the second last RBNE.
 2. Then wait one I2S clock cycle.
 3. Clear the I2SEN bit.

I2S slave transmission sequence

The transmission sequence in slave mode is similar to that in master mode. The differences between them are described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The transmission sequence begins when the external master sends the clock and when the I2S_WS signal requests the transfer of data. The data has to be written to the SPI_DATA register before the master initiates the communication. Software should write the next audio data into SPI_DATA register before the current data finishes. Otherwise, transmission underrun error occurs. The TXURERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is mandatory to disable and enable I2S to resume the communication. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

I2S slave reception sequence

The reception sequence in slave mode is similar to that in master mode. The differences between them are described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The reception sequence begins when the external master sends the clock and when the I2S_WS signal indicates a start of the data transfer. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit immediately after receiving the last RBNE.

I2S full-duplex mode

A single I2S only supports one-way transmission: transmit or receive mode. I2S full-duplex is supported by using an extra I2S module: I2S_ADD simultaneously with I2S. I2S_ADD module has the same function with I2S module, but can only work in slave mode. There is only one I2S_ADD1 module, so only I2S1 supports full-duplex mode. I2S_ADD's I2S_CK and I2S_WS are internally connected to its respective I2S's respective ports. I2S_ADD's I2S_SD pin is mapped to respective I2S's SPI_MISO pin.

In order to work in full-duplex mode, application should enable the I2S module as well as its corresponding I2S_ADD module. I2S supports two full-duplex modes: master mode and slave mode.

In master full-duplex mode, software should set I2S as a master, and I2S_ADD as a slave. Then I2S_ADD's WS and SCK signals come from the master I2S.

In slave full-duplex mode, software should set both I2S and I2S_ADD as slaves. Then, the WS and CK signals of both I2S_ADD and I2S come from external.

Application may configure I2S into either a transmitter or a receiver and thus, configure I2S_ADD into opposite data direction. During transmission, software should operate registers and handle interrupts for both I2S and I2S_ADD to make a full-duplex transmission.

20.9.4. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

20.10. I2S interrupts

20.10.1. Status flags

There are four status flags implemented in the SPI_STAT register, including TBE, RBNE, TRANS and I2SCH. The user can use them to fully monitor the state of the I2S bus.

- 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.

- I2S 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 will not generate any interrupt.

- I2S channel side flag (I2SCH)

This flag indicates the channel side information of the current transfer and has no meaning in PCM mode. It is updated when TBE rises in transmission mode or RBNE rises in reception mode. This flag will not generate any interrupt.

20.10.2. Error flags

There are three error flags:

- Transmission underrun error flag (TXURERR)

This situation occurs when the transmit buffer is empty if the valid SCK signal starts in slave transmission mode.

- Reception overrun error flag (RXORERR)

This situation occurs when the receive buffer is full and a newly incoming data has been completely received. When overrun occurs, the data in receive buffer is not updated and the newly incoming data is lost.

- Format Error (FERR)

In slave I2S mode, the I2S monitors the I2S_WS signal and an error flag will be set if I2S_WS toggles at an unexpected position.

I2S interrupt events and corresponding enable bits are summed up in the [Table 20-8. I2S interrupt](#).

Table 20-8. I2S interrupt

Interrupt 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
TXURERR	Transmission underrun error	Read SPI_STAT register	ERRIE
RXORERR	Reception overrun error	Read SPI_DATA register and then read SPI_STAT register.	
FERR	I2S format error	Read SPI_STAT register	

20.11. Register definition

SPI0 secure access base address: 0x5001 3000
 SPI0 non-secure access base address: 0x4001 3000
 SPI1/I2S1 secure access base address: 0x5000 3800
 SPI1/I2S1 non-secure access base address: 0x4000 3800
 I2S1_add secure access base address: 0x5000 3400
 I2S1_add non-secure access base address: 0x4000 3400

20.11.1. 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).

This register has no meaning in I2S mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC[2:0]	MSTMOD	CKPL	CKPH		

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 information transfers between the MOSI pin in master and the MISO pin in 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: CRC calculation is disabled 1: CRC calculation is enabled
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value (TCRC)

		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 mode 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: SPI peripheral is disabled 1: SPI peripheral is enabled
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/2 100: PCLK/32 001: PCLK/4 101: PCLK/64 010: PCLK/8 110: PCLK/128 011: PCLK/16 111: PCLK/256 PCLK means PCLK2 when using SPI0 or PCLK1 when using SPI1
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
---	------	--

20.11.2. 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).

31	30	29	28	27	26	25	24	23	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		TBEIE	RBNEIE	ERRIE	TMOD	Reserved	NSSDRV	DMATEN	DMAREN	

rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: TBE interrupt is disabled. 1: TBE interrupt is enabled. An interrupt is generated when the TBE bit is set.
6	RBNEIE	Receive buffer not empty interrupt enable 0: RBNE interrupt is disabled. 1: RBNE interrupt is enabled. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable 0: Error interrupt is disabled. 1: Error interrupt is enabled. 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: SPI TI mode disabled. 1: SPI TI mode enabled.
3	Reserved	Must be kept at reset value.
2	NSSDRV	Drive NSS output 0: NSS output is disabled. 1: NSS output is enabled. If the NSS pin is configured as output, the NSS pin is pulled low in master mode when SPI is enabled. If the NSS pin is configured as input, the NSS pin should be pulled high in master mode, and this bit has no effect.
1	DMATEN	Transmit buffer DMA enable

0: Transmit buffer DMA is disabled.
 1: Transmit buffer DMA is enabled, when the TBE bit in SPI_STAT is set, there will be a DMA request on corresponding DMA channel.

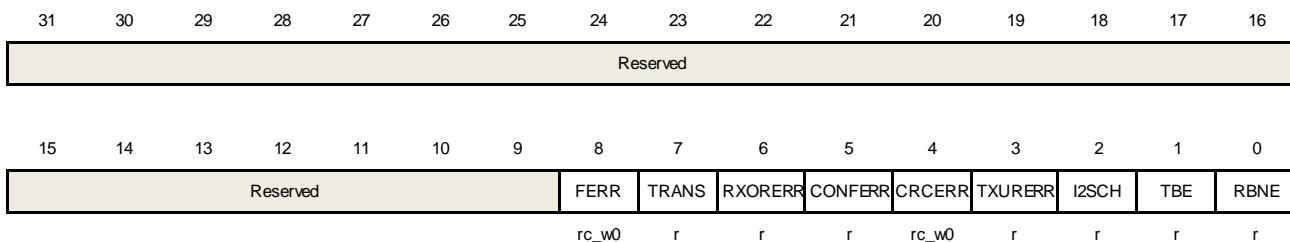
0	DMAREN	Receive buffer DMA enable 0: Receive buffer DMA is disabled. 1: Receive buffer DMA is enabled, when the RBNE bit in SPI_STAT is set, there will be a DMA request on corresponding DMA channel.
---	--------	--

20.11.3. 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 I2S Mode: 0: No I2S format error 1: I2S format error occurs This bit is set by hardware and cleared by writing 0.
7	TRANS	Transmitting ongoing bit 0: SPI or I2S is idle. 1: SPI or I2S 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. This bit is not used in I2S mode.
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. This bit is not used in I2S mode.
3	TXURERR	Transmission underrun error bit 0: No transmission underrun error occurs. 1: Transmission underrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_STAT register. This bit is not used in SPI mode.
2	I2SCH	I2S channel side 0: The next data needs to be transmitted or the data just received is channel left. 1: The next data needs to be transmitted or the data just received is channel right. This bit is set and cleared by hardware. This bit is not used in SPI mode, and has no meaning in the I2S PCM mode.
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

20.11.4. 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
SPI_DATA[15:0]															

rw

Bits	Fields	Descriptions
------	--------	--------------

31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	<p>Data transfer register</p> <p>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.</p> <p>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.</p>

20.11.5. 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).

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	<p>CRC polynomial value</p> <p>These bits contain the CRC polynomial and they are used for CRC calculation. The default value is 0007h.</p>

20.11.6. 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).

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

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	<p>RX CRC value</p> <p>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].</p> <p>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.</p> <p>This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.</p>

20.11.7. 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).

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

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	<p>TX CRC value</p> <p>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].</p> <p>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.</p> <p>This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.</p>

20.11.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C

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	I2SEL	I2SEN	I2SOPMOD[1:0]	PCMSMO D	Reserved	I2STD[1:0]	CKPL	DTLEN[1:0]	CHLEN						

rw rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	I2SEL	I2S mode selection 0: SPI mode 1: I2S mode This bit should be configured when SPI/I2S is disabled.
10	I2SEN	I2S enable 0: I2S is disabled 1: I2S is enabled This bit is not used in SPI mode.
9:8	I2SOPMOD[1:0]	I2S operation mode 00: Slave transmission mode 01: Slave reception mode 10: Master transmission mode 11: Master reception mode This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
7	PCMSMOD	PCM frame synchronization mode 0: Short frame synchronization 1: Long frame synchronization This bit has a meaning only when PCM standard is used. This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
6	Reserved	Must be kept at reset value.
5:4	I2STD[1:0]	I2S standard selection 00: I2S Phillips standard 01: MSB justified standard

		10: LSB justified standard 11: PCM standard These bits should be configured when I2S is disabled. These bits are not used in SPI mode.
3	CKPL	Idle state clock polarity 0: The idle state of I2S_CK is low level 1: The idle state of I2S_CK is high level This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
2:1	DTLEN[1:0]	Data length 00: 16 bits 01: 24 bits 10: 32 bits 11: Reserved These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
0	CHLEN	Channel length 0: 16 bits 1: 32 bits The channel length must be equal to or greater than the data length. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

20.11.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

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					MCKOEN	OF	DIV[7:0]								
					rw	rw	rw								

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled 1: I2S_MCK output is enabled This bit should be configured when I2S is disabled.

		This bit is not used in SPI mode.
8	OF	Odd factor for the prescaler 0: Real divider value is DIV * 2 1: Real divider value is DIV * 2 + 1 This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
7:0	DIV[7:0]	Dividing factor for the prescaler Real divider value is DIV * 2 + OF. DIV must not be 0. These bits should be configured when I2S is disabled. These bits are not used in SPI mode.

20.11.10. Quad-SPI mode control register (SPI_QCTL) of SPI0

Address offset: 0x80

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												IO23_DR V			QRD	QMOD	
															rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	IO23_DRV	Drive IO2 and IO3 enable 0: IO2 and IO3 are not driven in single wire mode 1: IO2 and IO3 are driven to high in single wire mode This bit is only available in SPI0.
1	QRD	Quad-SPI mode read select 0: SPI is in quad wire write mode 1: SPI is in quad wire read mode This bit should be only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI0.
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). This bit is only available in SPI0.

21. Serial/Quad Parallel Interface (SQPI)

21.1. Overview

Serial/Quad Parallel Interface (SQPI) is a controller for external serial/dual/quad parallel interface memory peripheral. For example: SQPI-PSRAM and SQPI-FLASH.

With this controller, users can use external SQPI interface memory as SRAM simply.

21.2. Characteristics

- SQPI controller has two independent sets of configure registers for write operation and read operation.
- SQPI controller support ID length setting.
- SQPI controller can configure the sampling edge of the SQPI_CLK during the read operation.
- SQPI controller support configuring the length of command phase, address phase, and waitcycle phase.
- SQPI controller support configuring output clock frequency which is divided by HCLK.
- SQPI controller support no address phase and data phase operation which is named special command by the controller.
- SQPI controller support READ ID command which is more than 32 bit data during one AHB command.
- SQPI controller support AHB burst operation and 8/16/32 bit AHB command.
- SQPI controller support 256MB external memory space.
Logic memory address range: 0x6000_0000 - 0x6FFF_FFFF.
- SQPI controller support 6 types mode for different combination of command, address, waitcycle, and data phase.

21.3. Function overview

21.3.1. SQPI mode definition

In mode name, the first character indicates command phase valid IO number, the second indicates address phase valid IO number, and the third indicates data phase valid IO number. For each character, S means single (1 IO), D means dual (2 IO), Q means quad (4 IO)

Table 21-1. SQPI controller mode definition

Signal	Direction	Operation Mode					
		SSS	SSQ	SQQ	QQQ	SSD	SDD
SQPI_CLK	Output			Serial Clock			

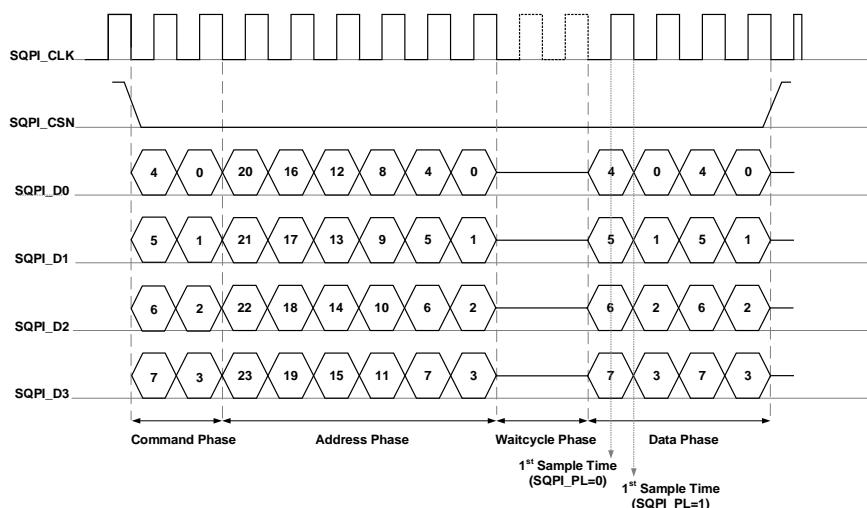
SQPI_CSN	Output	Chip-Enable (active low)					
Command Phase							
SQPI_D0	Output	O	O	O	O	O	O
SQPI_D1	Output	X	X	X	O	X	X
SQPI_D2	Output	0	0	0	O	0	0
SQPI_D3	Output	1	1	1	O	1	1
Address Phase							
SQPI_D0	Output	O	O	O	O	O	O
SQPI_D1	Output	X	X	O	O	X	O
SQPI_D2	Output	0	0	O	O	0	0
SQPI_D3	Output	1	1	O	O	1	1
Waitcycle Phase							
SQPI_D0	Inout	X	X	X	X	X	X
SQPI_D1	Inout	X	X	X	X	X	X
SQPI_D2	Inout	0	X	X	X	0	0
SQPI_D3	Inout	1	X	X	X	1	1
Data Phase							
SQPI_D0	Inout	O	IO	IO	IO	IO	IO
SQPI_D1	Inout	I	IO	IO	IO	IO	IO
SQPI_D2	Inout	X	IO	IO	IO	X	X
SQPI_D3	Inout	X	IO	IO	IO	X	X

Note: O – Output, I – Input, IO – Inout, 0 – Output 0, 1 – Output 1, X - Hiz

21.3.2. SQPI controller sampling polarity

SQPI controller read operation sampling polarity (PL bit in SQPI_INIT register) selection function support user to change the controller sampling time. This function is highly useful when SQPI clock is high. Example showed as below:

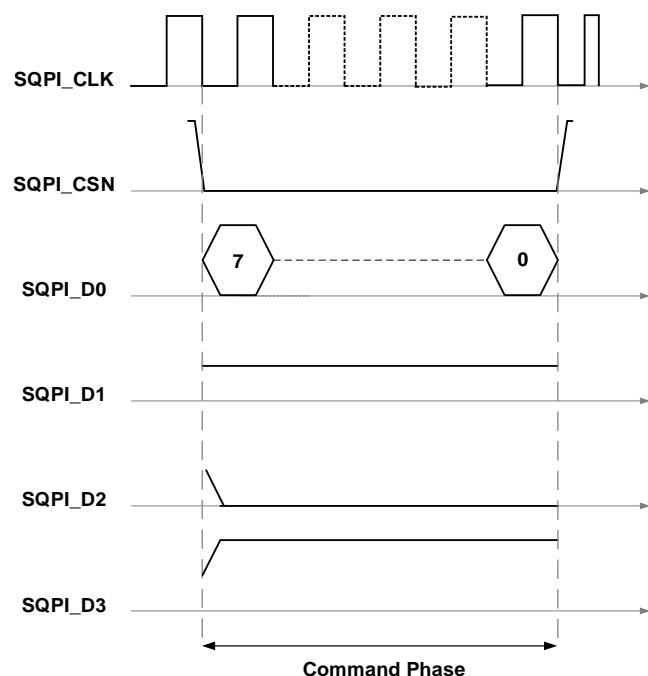
Figure 21-1. SQPI Polarity Example



21.3.3. SQPI controller special command

SQPI controller special command (SCMD bit in SQPI_WCMD register) function can send only command phase with no address, waitcycle, and data phase. Special command function will be mandatory to SSS mode by hardware. If you set SCMD bit to 1, you must read this bit and wait it cleared before doing other memory access because this can ensure the operation has performed in the interface.

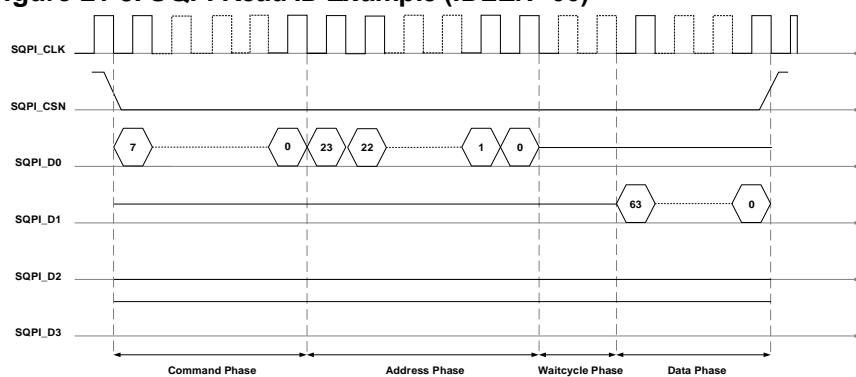
Figure 21-2. SQPI SCMD Example



21.3.4. SQPI controller read ID command

For more than 32-bit ID data, RDID function can supply help. To use this function, first you should set IDLEN bit(SQPI_INIT register) to 0x00(64bit, this is default), then set the RDID(SQPI_RCMD register) bit to 1 and wait it cleared by hardware through polling this bit, and at last read the IDL and IDH registers. This command is performed in SSS mode by hardware.

Figure 21-3. SQPI Read ID Example (IDLEN=00)



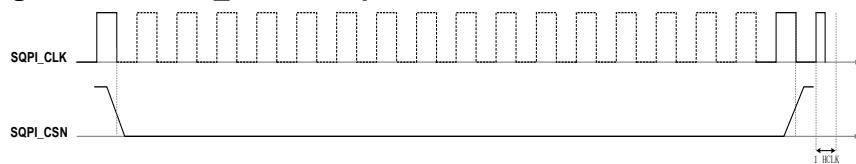
21.3.5. SQPI controller output clock configuration

SQPI clock period is configured by CLKDIV bits(SQPI_INIT register). The frequency formula of SQPI_CLK is:

$$f_{sqpi_clk} = \frac{f_{hclk}}{CLKDIV + 1}$$

Note: CLKDIV cannot be 0. When CLKDIV field is even number, the output clock high level time has 1 HCLK period more than low level time. After the rise edge of SQPI_CSN there is 1 HCLK period clock to on SQPI_CLK signal to support some old PSRAM memory.

Figure 21-4. SQPI_CLK Example



21.3.6. SQPI controller initialization

In the beginning, users should program the initialization register SQPI_INIT. Data sampling clock edge is selected via the PL bit, read device ID length could be configured by the IDLEN bits, address bit number is controlled by the ADDRBIT, command bit number is set by CMDBIT, and the SQPI controller clock is configured by CLKDIV bits.

21.3.7. Read ID command flow

The first, user should configure RCMD bits by Read ID command (e.g. 0x9F for SQPIPSRAM) and read waitcycle number in SQPI_RCMD register. The second user sets RID bit to 1 and wait it reset to 0. The third, user can get ID value by read SQPI_IDL and SQPI_IDH registers.

21.3.8. Read/Write operation flow

Six modes of memory access are possible. Access mode should be configured before read/write operations. Read/Write command mode is programmed by the RMODE and WMODE, wait cycle is controlled by the RWAITCYCLE and WWAITCYCLE bit, and the specific memory operating command should be programmed in RCMD and WCMD bit, these read/write settings are located in SQPI_RCMD and SQPI_WCMD registers respectively.

After memory access mode configuration, user can directly access external device as SRAM by using SQPI memory logic address.

21.3.9. SQPI controller mode timing

SQPI controller mode timing for read/write operation, each AHB read/write access to SQPI memory logic address will transfer to one of below timing:

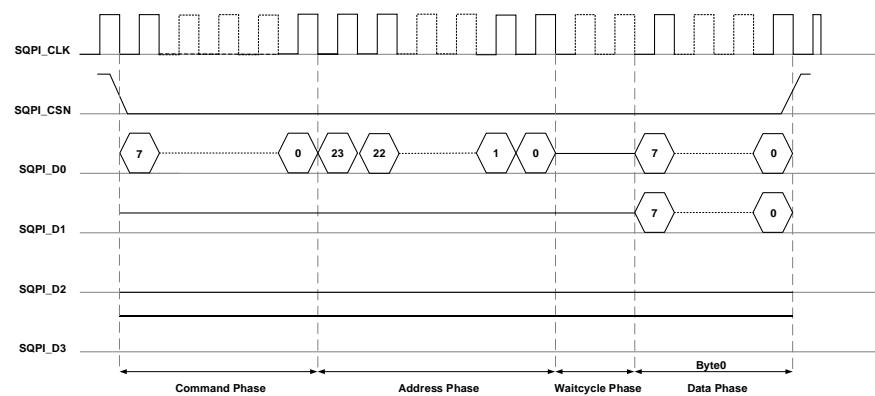
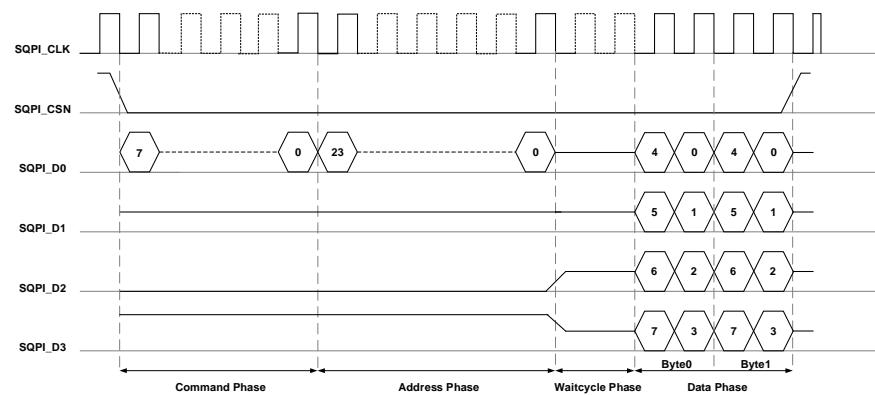
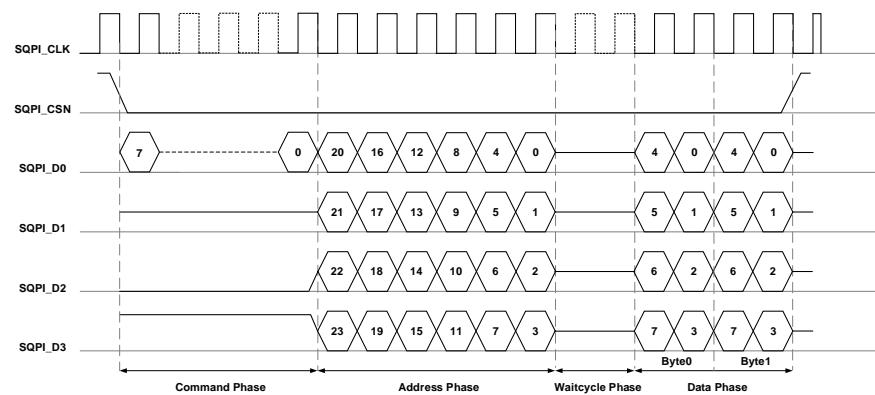
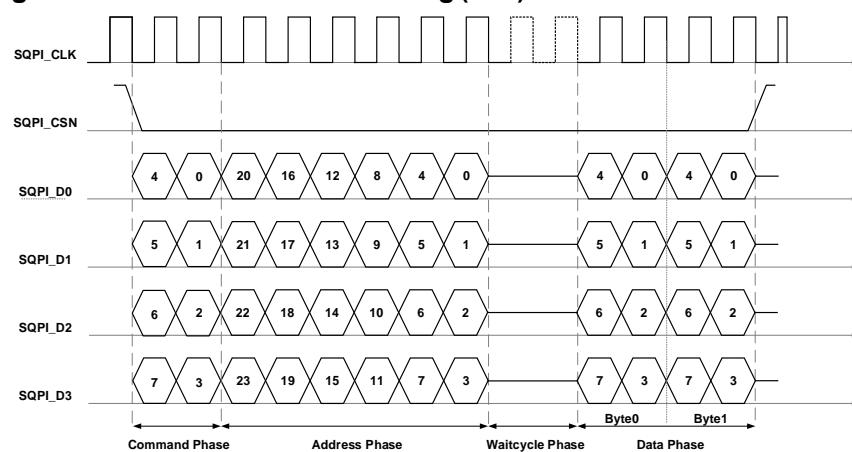
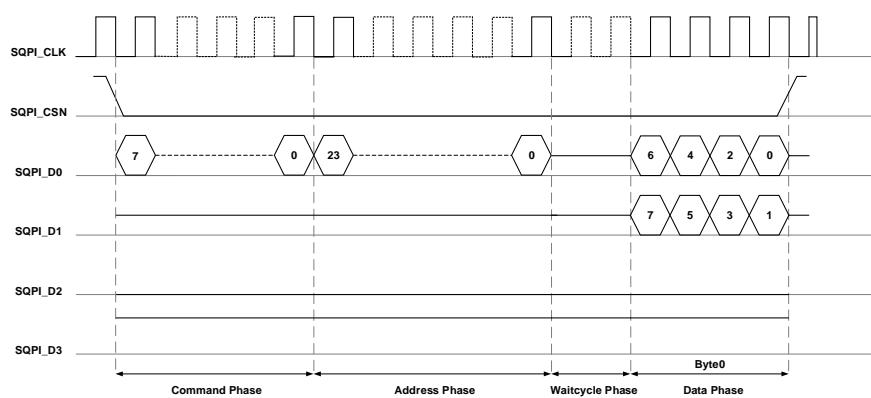
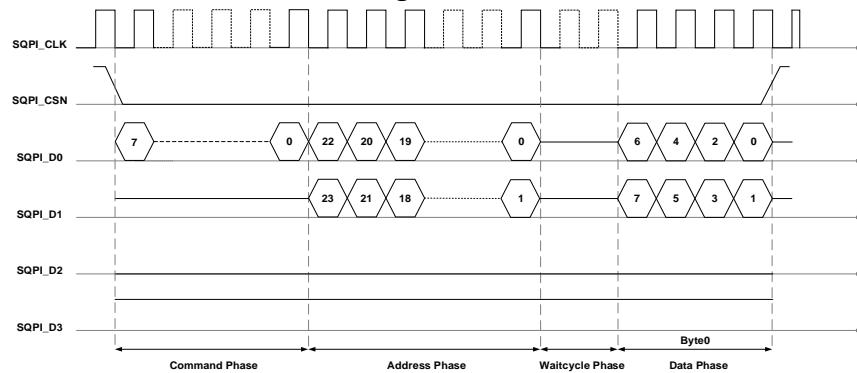
Figure 21-5. SQPI SSS Mode Timing (SPI)

Figure 21-6. SQPI SSQ Mode Timing

Figure 21-7. SQPI SQQ Mode Timing (SQPI)


Figure 21-8. SQPI QQQ Mode Timing (QPI)**Figure 21-9. SQPI SSD Mode Timing****Figure 21-10. SQPI SDD Mode Timing**

21.4. Register definition

SQPI secure access base address: 0x5002 5400

SQPI non-Secure access base address: 0x4002 5400

21.4.1. SQPI Initial Register (SQPI_INIT)

Address offset: 0x00

Reset Value: 0x1801 0004

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
PL	IDLEN[1:0]		ADDRBIT[4:0]				CLKDIV[5:0]				CMDBIT[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
Reserved															

Bits	Fields	Descriptions
31	PL	Read data sample polarity. 0: Sample data at rising edge(default) 1: Sample data at falling edge.
30:29	IDLEN[1:0]	SQPI controller external memory ID length. 00:64-bit 01:32-bit 10:16-bit 11:8-bit
28:24	ADDRBIT[4:0]	Bit number of SPI PSRAM address phase. Default: 24
23:18	CLKDIV[5:0]	Clock divider for SQPI output clock. 0x0 is invalid. Output clock frequency is $f_{hclk}/(CLKDIV+1)$ Note: When CLKDIV field is even number, the output clock high level time has 1 HCLK period more than low level time.
17:16	CMDBIT[1:0]	Bit number of SQPI controller command phase 00: 4 bit 01: 8 bit (default) 10: 16 bit 11: Reserved
15:0	Reserved	Must be kept at reset value.

21.4.2. SQPI Read Command Register (SQPI_RCMD)

Address offset: 0x04

Reset value: 0x0010 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
RID	Reserved								RMODE[2:0]		RWAITCYCLE[3:0]				
rw									rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCMD[15:0]															
rw															

Bits	Fields	Descriptions
31	RID	Send read ID command, command code comes from RCMD.
30:23	Reserved	Must be kept at reset value.
22:20	RMODE[2:0]	SQPI controller read command mode: 000: SSQ mode 001: SSS mode 010: SQQ mode 011: QQQ mode 100: SSD mode 101: SDD mode
19:16	RWAITCYCLE[3:0]	SQPI read command waitcycle number after address phase
15:0	RCMD[15:0]	SQPI read command for AHB read transfer RCMD[3:0] are valid when CMDBIT=00 RCMD[7:0] are valid when CMDBIT=01 RCMD[15:0] are valid when CMDBIT=10

NOTE: Before write 1 to RID bit, you must ensure it is cleared and after set RID to 1, you must wait RID cleared

21.4.3. SQPI Write Command Register (SQPI_WCMD)

Address offset: 0x08

Reset value: 0x0001 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
SCMD	Reserved								WMODE [2:0]		WWAITCYCLE[3:0]				
rs									rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCMD[15:0]															
rw															

Bits	Fields	Descriptions
31	SCMD	Send special command which does not have address and data phase, command

		code comes from WCMD.
30:23	Reserved	Must be kept at reset value.
22:20	WMODE[2:0]	SQPI controller write command mode: 000: SSQ mode 001: SSS mode 010: SQQ mode 011: QQQ mode 100: SSD mode 101: SDD mode
19:16	WWAITCYCLE[3:0]	SQPI write command waitcycle number after address phase
15:0	WCMD[15:0]	SQPI write command for AHB write transfer

Note : Before write 1 to SC bit, you must ensure it is cleared and after set SC to 1, you must wait SC cleared

21.4.4. SQPI ID Low Register (SQPI_IDL)

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
IDL [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDL [15:0]															

Bits	Fields	Descriptions
31:0	IDL[31:0]	ID Low Data saved for SQPI Read ID Command IDL[15:0] is valid when IDLEN=10 IDL[7:0] is valid when IDLEN=11.

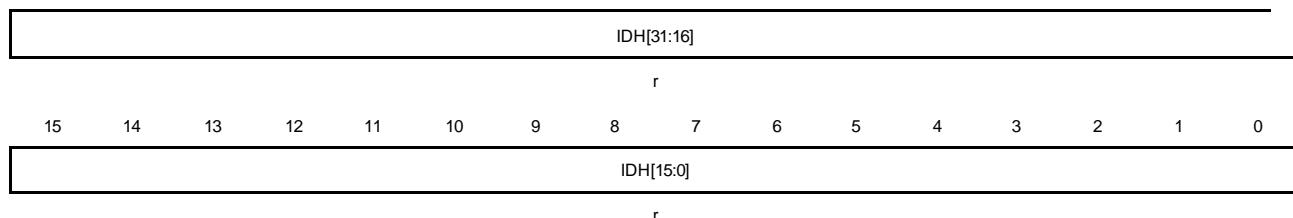
21.4.5. SQPI ID High Register (SQPI_IDH)

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
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Bits	Fields	Descriptions
31:0	IDH[31:0]	ID High Data saved for SQPI read ID command This register only valid when IDLEN = 00.

22. Quad-SPI interface (QSPI)

22.1. Overview

The QSPI is a specialized interface that communicate with Flash memories. This interface support single, dual or quad SPI FLASH. It can operate in any of the 4 following modes:

- indirect mode(address extend): all operations are performed depends on QSPI registers
- status polling mode: the values of status registers in external Flash memory are periodically read and check
- memory-mapped mode: the external Flash memory is mapped to the microcontroller address space(0x9000 0000 - 0x97FF FFFF) and is accessed as an internal memory
- FMC mode: the external Flash memory(0x0800 0000/0x0C00 0000) is accessed as an internal memory by flash controller for extensions

22.2. Characteristics

- Four functional modes: indirect(address extend), status-pollling, memory-mapped and FMC mode
- Fully programmable command format for both indirect and memory mapped mode
- Integrated FIFO for transmission/reception
- 8, 16, or 32-bit data accesses
- DMA channel for indirect mode
- Interrupt generation on FIFO threshold, status match, timeout, transfer complete, and access error
- Support TrustZone architecture to isolate the secure area and non-secure area

22.3. Function overview

22.3.1. QSPI block diagram

Module QSPI mainly has a connection with TZPCU, AHB, FMC and PAD. 6 signals are used to interface with an external flash memory,The specific signal description is showed as follow table.

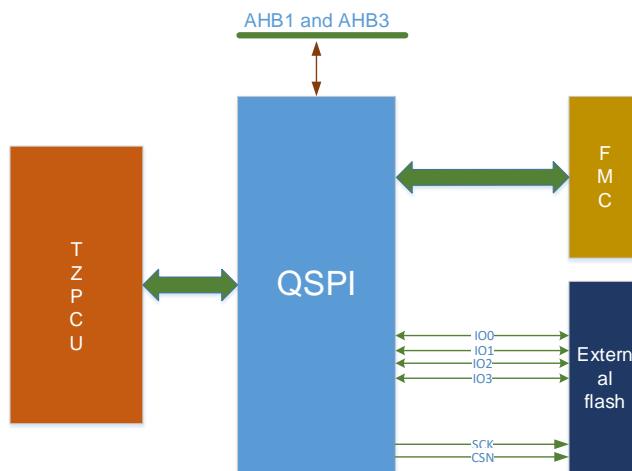
TZPCU provides essential information related to trustzone for QSPI. QSPI TrustZone mode can be configured TrustZone -aware or securable. In TrustZone -aware mode, QSPI will get TZEN and watermarked address range through TZPCU. In securable mode, QSPI will get TZEN, watermarked address range and security properties through TZPCU.

The connection between QSPI and FMC is used in QSPI FMC mode. QSPI can accomplish program start with external flash through FMC interface.

Table 22-1. QSPI signal description

Pin name	Direction	Description
CSN	O	chip select output (active low)
SCK	O	clock output
IO0/SO	I/O	single mode: data output dual mode: data input or output quad mode: data input or output
IO1/SI	I/O	single mode: data input dual mode: data input or output quad mode: data input or output
IO2	I/O	single mode: connect WP pin of flash, control "write protect" function dual mode: connect WP pin of flash, control "write protect" function quad mode: data input or output
IO3	I/O	single mode: connect HOLD pin of flash, control "hold" function dual mode: connect HOLD pin of flash, control "hold" function quad mode: data input or output

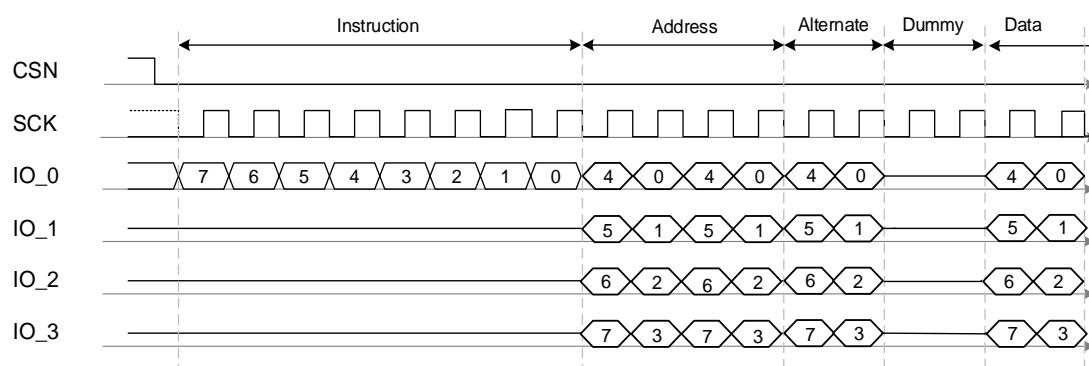
[Figure 22-1. QSPI diagram](#) shows the block diagram of the QSPI unit.

Figure 22-1. QSPI diagram


22.3.2. QSPI command format

The QSPI communicates with the Flash memory using commands in various formats. There are totally 5 phases which can be included or not: instruction, address, alternate byte, dummy and data. Any of these phases can be configured to be omitted or not, but at least one of the instructions, address, alternate byte, or data phase must be present, this must be guaranteed by software, hardware is not designed to provide any protection methods. In addition, the most-significant-bit always occupies the highest IO line number.

Figure 22-2. QSPI command format



Instruction phase

In this phase, the 8-bit instruction, configured in INSTRUCTION field (QSPI_TCFG register) is sent to the flash memory.

IMOD field (QSPI_TCFG register) defines the instruction phase mode (no instruction, 1-line, 2-lines, or 4-lines).

Address phase

In this phase, 1-4 bytes of address are sent to the flash memory.

In Indirect mode, ADDR field (QSPI_ADDR register) defines the information of address. ADDRSZ field (QSPI_TCFG register) defines the number of address byte to be sent.

ADDRMOD field (QSPI_TCFG register) defines the address phase mode (no address, 1-line, 2-lines, or 4-lines).

When QSPI FMC mode use and TZEN is set, in this phase QSPI also judges if current indirect address with CPU secure working status is an accessible transfer. If not qspi will generate an interrupt and set a transfer error flag.

Alternate-bytes phase

In this phase, 1-4 alternate-bytes are sent to the flash memory.

ALTE field (QSPI_ALTE register) defines the information of alternate bytes, ALTESZ field (QSPI_TCFG register) defines the number of alternate-bytes to be sent, ALTEMOD field

(QSPI_TCFG register) defines the alternate bytes phase mode (no alternate bytes, 1-line, 2-lines, or 4-lines).

Dummy phase

In this phase, 0-31 cycles, as specified by DUMYC field (QSPI_TCFG register), are given without any data being transferred for external flash, in order to wait flash prepare data.

DATAMOD field (QSPI_TCFG register) defines the dummy phase mode (1-line, 2-lines, or which is used in Data phase.

Data phase

In this phase, any number of bytes can be transferred between the external flash memory and the QSPI interface.

In indirect mode, DTLEN field (QSPI_DTLEN register) defines the number of bytes to be sent/received. In write operation, data to be sent should be written to the DATA register, while in read operation, received data is obtained by reading DATA register.

In memory-mapped mode, the number of bytes to be transmitted is specified as single AHB bus access operation, these could be 8, 16 or 32 read/write access, corresponding to 1, 2, or 4 bytes. Also when the TZEN is set, QSPI in memory-mapped mode will check that if the haddr with the CPU secure status is an accessible transfer. If not, qspi will generate a hard-fault.

DATAMOD filed (QSPI_TCFG register) defines the data phrase mode (no data, 1-line, 2-lines, or 4-lines), and the configuration of DATAMOD = 00 must only be used in indirect write mode.

22.3.3. QSPI signal line modes

Each of the instruction, address, alternate-byte, or data phase can be configured separately into signal line modes by setting IMOD/ ADDRMOD/ ALTEMOD/ DATAMOD

Table 22-2. QSPI singnal line modes

Signal line modes		Single mode	Dual mode	Quad mode
Config filed	IMOD	01 or 00	10 or 00	11 or 00
	ADDRMOD			
	ALTEMOD			
	DATAMOD			

Signal line modes		Single mode	Dual mode	Quad mode	
Pins	IO0 (SO)	Output	Input: data read (high impedance) output: all other phases	Input : data read (high impedance) Output: all other phases.	
	IO1 (SI)	Input (high impedance)			
	IO2	Output 0 (deactivate "write protect function")			
	IO3	Output 1 (deactivate "hold" function)			
Description		In dummy phase when DATAMOD = 2'b01, IO0 output, IO1 input (high impedance)	In dummy phase when DATAMOD = 2'b10, IO0/IO1 are always high-impedance.	In dummy phase when DATAMOD = 2'b11, IO0/IO1/IO2/IO3 are always high-impedance.	

IO2/IO3 are used only in quad mode, if none of the 5 phases are configured in quad mode, then IO2/IO3 are released and can be used for other functions even when QSPI is enabled.

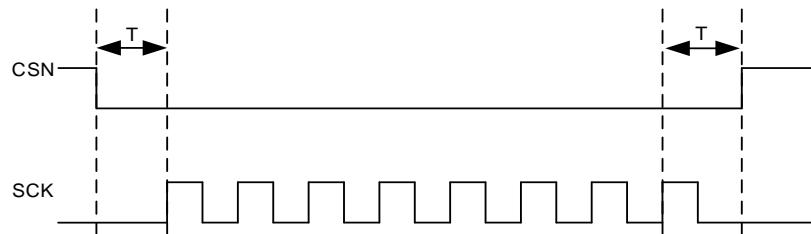
22.3.4. CSN and SCK behavior

The default value of CSN is high, and it falls before a command begins and rises as soon as it finishes.

SCK output signal is a gate signal from internal sck, where the internal sck is present all the time.

CSN falls one SCK cycle before the first valid rising SCK edge, and rises on SCK cycle after the final valid rising SCK edge.

Figure 22-3. CSN and SCK behavior



When the FIFO stays empty in a write command, or full in a read command, SCK will be stalled and stays low until the FIFO can work again. At this moment if the CSN is high, SCK will rise back up one half of a SCK cycle after the rising edge of the CSN.

22.4. Operating modes

22.4.1. Indirect mode

In indirect write mode, data to be transmitted are written into DATA. While in indirect read mode, data to be received are read from DATA.

DTLEN field (QSPI_DTLEN register) defines the number of byte to be transferred. If DTLEN = 0xFFFF_FFFF, the number of data is considered undefined, the transmission continues until the memory size boundary is reached as specified by FMSZ. If both DTLEN = 0xFFFF_FFFF and FMSZ = 0x1F, then the transmission continues indefinitely until the QSPI is disabled.

Transfer complete flag TC is set when the number of byte programmed in DTLEN is reached, in case of undefined transfer length, TC is set when the transmit/received byte number equals to external memory size. An interrupt is generated if TCIE and TC are both set, and it is cleared by setting TCC to 1.

Trigger a command sequence

The command sequence starts immediately after the last information is provided by software according to communication requirement.

When neither address nor data are required, the sequence starts immediately after TCFG has been accessed.

When address is required and no data is required, the sequence starts after ADDR has been accessed.

When both address and data are required in indirect write mode, the command sequence starts after DATA has been accessed.

FIFO and flag control

A FIFO with a size of 8-bit by 16 is implemented to transfer data. In indirect write mode, 32-bit AHB write access add 4-bytes to FIFO, 16-bit add 2-bytes, and 8-bit add 1-byte.

FIFO threshold is defined by FTL, in indirect read mode, when the amount of bytes in the FIFO is equal or above the defined threshold, FIFO threshold flag FT is set. FT is also set after data phase is complete if FIFO is not empty. In indirect write mode, when the amount of the empty bytes in the FIFO is above the threshold, FT is set.

An interrupt is generated if both FTIE and FT is set. If DMA is enabled, a DMA request is generated by FT, until this flag is cleared.

In indirect read mode, when the FIFO becomes full, the QSPI temporarily stop SCK clock to avoid overrun. The reading sequence is not resumed until more than 4 byte are available in FIFO.

22.4.2. Status polling mode

In status polling mode, the QSPI periodically starts a read command with up-to 4-bytes data. The received data can be bit-wise masked and compared with a defined data content, if a match happens, then an interrupt is generated when SMIE is set.

Status polling access starts the same as indirect read sequence. BUSY stays high even between periodic intervals.

Polling match mode SPMOD controls the comparison match mode, if SPMOD = 1, the AND mode is selected. In this mode, status match flag SM is set only when there is a match on all the unmasked bits. While if SPMOD = 0, the OR mode is selected. In this mode, SM is set if there is a match on any of the unmasked bit.

If status-polling-mode-stop SPS is set, status polling sequence stop when a match is detected, and the BUSY flag is cleared at the end of data phase. Otherwise, the periodic sequence continues until abort is issued or the QSPI is disabled.

In status polling mode, FIFO is bypassed, the read status bytes are stored in DATA, and the stored status bytes are not affected by the MASK control field. DATA contents is renewed at the beginning of data phase if there is any.

FT is set at the end of data phase, where the external flash memory status bytes are considered read, and it is cleared when DATA is read.

22.4.3. Memory map mode

In memory-mapped mode, the external flash memory is considered as internal memory, no more than 256MB can be address even if the external memory is larger. The Memory map mode also don't allow an address outside what defined by FMSZ but still within 256MB range. The memory map mode supports TrustZone architecture by stopping the transfer

which is judge as an unaccessible transfer(such as a secure transfer tries to access a nonsecure area), and generate a hard-fault.

If any of above condition happens, AHB will generate an error .The effect of the error depends on the AHB master.

In this mode, byte, half-word, and word single or burst access are supported.

CPU: a hard fault is generated

DMA: transfer error is generated, and the corresponding DMA channel is disabled.

Execute in place (XIP) is also supported, where the QSPI anticipates the next MCU access and load in advance the byte at the following address, if the subsequent access is indeed made at a continuous address, the access will be completed faster since the value is already prefetched. Otherwise, the read sequence is restarted, polling CSN low before the read sequence starts.

After the FIFO is full, the QSPI enters hold state, in which no SCK is sent, CSN is maintained low during this period. If timeout counter is enabled, CSN will be pulled high when hold state contains number of SCK clock cycles equals to TMOUT control field.

At the beginning of a transfer, BUSY goes high before CSN falls, and is cleared when a timeout occurred or abort/disable is issued.

22.4.4. FMC mode

This mode is supported based on memory map mode with a highest priority, QSPI read address is word align. FMC mode will abort any transfer except status polling mode. But when an indirect write mode is aborted, QSPI will not access FMC mode immediately because of the time requirement of external flash. As for staus polling mode, in order not to block the FMC mode, add timeout of staus polling.

NOTE: this mode will abort the normal transfer, and user should check the QSPI_BYTE_CNT register to know how many bytes are aborted. In this mode, ABORT bit in QSPI_CTL bit should be set to 1 to make sure that the data in FIFO will be stored until they are pushed or popped by software.

This mode will abort indirect mode and status mode. But when last mode is indirect write mode, FMC mode won't start right after indirect write mode is aborted or indirect write mode is finished, FMC mode will start after WTCNT which is specified by register QSPI_WTCNT.

When FMC mode is set to abort a staus-polling transfer, status-polling will still to work to match status until the time out flag for staus-polling mode which is specified by register QSPI_SPTMOUT, a hard fault is generated.

22.5. QSPI configuration

22.5.1. Flash configuration

The configuration in QSPI_DCFG register can be used to specify the characteristics of the external flash memory, so that the QSPI interface can work consistently.

FMSZ field defines the size of the external memory, FMSZ + 1 is the number of address bits in the flash memory. The maximum of the flash capacity can be up to 4GB in indirect mode.

CSHC field defines the chip select high time, it specify the minimum number of SCK cycles that CSN must stay high between two command sequences.

22.5.2. QSPI IP configuration

The configuration in QSPI_TCFG register can be used to specify the characteristics of the QSPI IP.

PSC field indicate the clock prescaler division factor.

SSAMPLE indicate which SCK edge is used to sample data. By default, the QSPI sample data one half of a SCK cycle after the external flash drives. However, it may be beneficial to sample data later because of the external signal delays. The sample edge can be shifted half one of SCK cycle using SSAMPLE bit.

DMAEN bit enables the DMA requests, which is generated according to FIFO level and FTL bits.

22.6. Security description

If there are option bytes, the global TrustZone system security is activated by setting the TZEN bit in FMC_OBR register. If there are no option bytes, the global TrustZone system security is activated by setting the TZEN bit in EFUSE_TZCTL register.

FMC QSPI mode read by FMC (FMC is the bus interface of the flash memory controller) or QSPI interface and program/erase by QSPI interface. Firstly it will judge the AHB addr[31:28], if addr[31:28] is 4 or 5, then the address will be a register address, otherwise the address will be a memory address when it is 9. Then if the address belongs to the register address range, QSPI will distinguish the register direction through the hnonsec signal and register property. Aimmg at the register address of QSPI_ADDR, QSPI will judge it's accessable through securemarked area partition.

If the address belongs to the memory address range, then QSPI directly judge the accessable of AHB addr through securemarked area partition.

Table 22-3. Flash secure/non-secure and privileged/unprivileged operation under

FMC mode when TrustZone is active (TZEN=1)

access type			non-secure register		secure register	
			PRIV=1	PRIV=0	PRIV=1	PRIV=0
read/w rite	secure operation	privileged accesses	OK		OK	
		unprivileged accesses	all read data is 0, w rite invalid	OK	all read data is 0, w rite invalid	OK
	non-secure operation	privileged accesses	OK		all read data is 0, w rite invalid, illegal access event	
		unprivileged accesses	all read data is 0, w rite invalid	OK		

22.7. Send instruction only once

Sending instruction only once is set by SIOO, this function is valid for all functional modes, if SIOO bit is set, the instruction is sent only once after QSPI_TCFG has been accessed. Subsequent command sequence skip instruction phase, until QSPI_TCFG is accessed again.

SIOO has no affect when IMOD = 00.

NOTE: Software should make sure that when use SIOO function, basic mode and FMC mode cannot be overlapped until one transfer is fully completed, or the result cannot be predicted.

22.8. Busy

BUSY bit is set once the QSPI start to operate the external flash memory.

In indirect mode, BUSY is reset once the command phase is end and if in indirect read mode, FIFO also needs be empty.

22.9. Error management

An error can be generated in the following case.

In indirect or status polling, TERR is generated immediately when a wrong address has been programmed in AR according to FMSZ.

In indirect mode, if the address (ADDR) plus data length (DTLEN) is greater than external memory size, TERR will be set once the QSPI is triggered.

In memory mapped mode, when an out of range access is done by AHB master or when the

QSPI is disabled, will generate an AHB error.

When an AHB master is accessing the memory mapped space while the memory mapped mode is not enabled, will generate an AHB error.

When wrong access operation (such as seqtrans attend to access the nonseq area)detected, support the error response to TZIAC, and if in memory map mode, will also return an AHB error.

When staus polling mode don't match until the timeout cnt for staus polling mode is deincreased to zero , will generate an AHB error.

22.10. QSPI interrupts

Table 22-4. QSPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
FT	FIFO threshold	By hardw are	FTIE
TC	Transfer complete	Set TCC bit in QSPI_STATC register	TCIE
TERR	Transfer error	Set TERRC bit in QSPI_STATC register	TERRIE
TMOUT	Timeout	Set TMOUTC bit in QSPI_STATC register	TMOUTIE
SM	Status match	Set SMC bit in QSPI_STATC register	SMIE
WS	Wrong start sequence	Set WSC bit in QSPI_STATC register	WSIE

22.11. Register definition

QSPI secure access base address: 0x5002 5800

QSPI non-secure access base address: 0x4002 5800

22.11.1. Control register (QSPI_CTL)

Address offset: 0x00

Reset value: 0x0000 0010

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	
PSC[7:0]							SPMOD	SPS	Reserved	FL[4:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FMCMOD	BUSY	Reserved	FTL[3:0]			Reserved	SCKDEN	SSAMPLE[1:0]	TMOUTE N	Reserved	ABORT	QSPIEN	w1s		rw	
r	r		rw			rw	rw	rw	rw	rw						

Bits	Fields	Descriptions
31:24	PSC [7:0]	<p>This field defines the scaler factor for generating SCK based on the AHB clock (value+1).</p> <p>0: $F_{CLK} = F_{AHB}$, 1: $F_{CLK} = F_{AHB} / 2$ 2: $F_{CLK} = F_{AHB} / 3$... 255: $F_{CLK} = F_{AHB} / 256$</p> <p>For odd clock division factors, CLK's duty cycle is not 50%. The clock signal remains low one cycle longer than it stays high.</p> <p>This field can be modified only when BUSY = 0.</p>
23	SPMOD	<p>Status Polling match mode</p> <p>0: AND match mode. SM is set if all the unmasked bits received from the Flash memory match the corresponding bits in the match register.</p> <p>1: OR match mode. SM is set if any one of the unmasked bits received from the Flash memory matches its corresponding bit in the match register.</p> <p>This bit can be modified only when BUSY = 0.</p>
22	SPS	<p>Status polling mode stop</p> <p>This bit determines if automatic polling is stopped after a match.</p> <p>0: Status polling mode is stopped only by abort or by disabling the QSPI.</p> <p>1: Status polling mode stops as soon as there is a match.</p> <p>This bit can be modified only when BUSY = 0.</p>

21	Reserved	Must be kept at reset value
20:16	FL[4:0]	FIFO level This field gives the number of valid bytes which are being held in the FIFO in indirect mode. In memory-mapped mode and in automatic status polling mode, FL is zero.
15	FMC_MOD	Busy in FMC mode This bit is set when a command is transferring in FMC mode. This bit is cleared once the operation with the Flash memory in FMC mode is completed.
14	BUSY	Busy This bit is set when a command is transferring. This bit is cleared once the operation with the Flash memory is finished and the FIFO is empty.
13:12	Reserved	Must be kept at reset value
11:8	FTL [3:0]	FIFO threshold level These bits are useful in indirect mode, the threshold number of bytes in the FIFO that will cause the FIFO threshold flag to be set. In indirect write mode (FMOD = 00): 0: FT is set if there are 1 or more free bytes available to be written to in the FIFO 1: FT is set if there are 2 or more free bytes available to be written to in the FIFO ... 15: FT is set if there are 16 free bytes available to be written to in the FIFO In indirect read mode (FMOD = 01): 0: FT is set if there are 1 or more valid bytes that can be read from the FIFO 1: FT is set if there are 2 or more valid bytes that can be read from the FIFO ... 15: FT is set if there are 16 valid bytes that can be read from the FIFO If DMAEN = 1, then the DMA controller for the corresponding channel must be disabled before changing the FTL value.
7	Reserved	Must be kept at reset value
6	SCKDEN	SCK delay enable when read data from flash, it is only useful when sample shift is 1 0: SCK delay disabled 1: SCK delay enabled
5:4	SSAMPLE[1:0]	Sample shift By default, the QSPI samples data 1/2 of a SCK cycle after the data is driven by the Flash memory. This bit allows the data to be sampled later in order to account for external signal delays. 0: No shift 1: 1/2 cycle shift 2: 1 cycle shift 3: Reserved

This field can be modified only when BUSY = 0.

3	TMOUTEN	Timeout counter enable
		This bit is valid only in memory-mapped mode (FMOD = 11). Activating this bit causes the chip select (CSN) to be released if there is no access after a certain amount of time, and this time is defined by TMOUT[15:0].
	0:	Timeout counter is disabled, and thus the chip select (CSN) remains active indefinitely after an access in memory-mapped mode.
	1:	Timeout counter is enabled, and thus the chip select is released in memory-mapped mode after TMOUT[15:0] cycles of Flash memory inactivity.
		This bit can be modified only when BUSY = 0.
2	Reserved	Must be kept at reset value
1	ABORT	Abort request
		This bit stops the current command. It is automatically cleared once the abort is complete.
		When FMC mode is used, this bit is set by hardware to stop the normal transfer and cleared when it is ready to work in FMC mode. User should not write this bit in this mode.
		In polling mode or memory-mapped mode, this bit also resets the SPS bit or the DMAEN bit.
	0:	No abort requested
	1:	Abort requested
0	QSPIEN	Enable
		Enable the QSPI.
	0:	QSPI is disabled
	1:	QSPI is enabled

22.11.2. Device configuration register (QSPI_DCFG)

Address offset: 0x04

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

Bits	Fields	Descriptions
------	--------	--------------

31:21	Reserved	Must be kept at reset value
20:16	FMSZ[4:0]	<p>Flash memory size</p> <p>This field defines the size of external memory using the following formula:</p> <p>Number of bytes in Flash memory = 2^{FMSZ+1}</p> <p>FMSZ+1 is effectively the number of address bits in the Flash memory. The Flash memory capacity can be up to 4GB in indirect mode, while it is limited to 256MB in memory mapped mode.</p> <p>This field can be modified only when BUSY = 0.</p>
15:11	Reserved	Must be kept at reset value
10:8	CSHC[2:0]	<p>Chip select high cycle</p> <p>CSHC+1 defines the minimum number of CLK cycles which the chip select(CSN) must stay high between two command sequences.</p> <p>0: CSN stays high for at least 1 cycle between Flash memory commands</p> <p>1: CSN stays high for at least 2 cycles between Flash memory commands</p> <p>...</p> <p>7: CSN stays high for at least 8 cycles between Flash memory commands</p> <p>This field can be modified only when BUSY = 0.</p>
7:1	Reserved	Must be kept at reset value
0	CKMOD	<p>This bit indicates the SCK level when QSPI is free</p> <p>0: CLK must stay low while CSN is high (QSPI is free).</p> <p>1: CLK must stay high while CSN is high (QSPI is free).</p> <p>This field can be modified only when BUSY = 0.</p>

22.11.3. Status register (QSPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0004

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	DMAEN	<p>DMA enable</p> <p>In indirect mode, DMA can be used to transfer data via QSPI_DATA. DMA transfers</p>

		are initiated when FT is set.
		0: DMA disabled 1: DMA enabled
15:12	Reserved	Must be kept at reset value
11	WSIE	Wrong start sequence interrupt enable This bit enables the wrong start sequence interrupt. 0: Interrupt disable 1: Interrupt enabled
10	TMOUTIE	Timeout interrupt enable This bit enables the timeout interrupt. 0: Interrupt disable 1: Interrupt enabled
9	SMIE	Status match interrupt enable This bit enables the status match interrupt. 0: Interrupt disable 1: Interrupt enabled
8	FTIE	FIFO threshold interrupt enable This bit enables the fifo threshold interrupt. 0: Interrupt disable 1: Interrupt enabled
7	TCIE	Transfer complete interrupt enable This bit enables the transfer complete interrupt. 0: Interrupt disable 1: Interrupt enabled
6	TERRIE	Transfer error interrupt enable This bit enables the transfer error interrupt. 0: Interrupt disable 1: Interrupt enabled
5	WS	Wrong start sequence flag This bit is set when a wrong secure start sequence is detected .This bit is cleared by writing 1 to WSC.
4	TMOUT	Timeout flag This bit is set when timeout occurs. It is cleared by writing 1 to TMOUTC.
3	SM	Status match flag This bit is set in status polling mode when the unmasked received data matches the expected value. It is cleared by writing 1 to SMC.
2	FT	FIFO threshold flag In indirect mode, this bit is set when the FIFO threshold has been reached, or if the FIFO is not empty after the last read operation from the Flash memory.

In automatic polling mode this bit is set every time the status register is read from the flash, and it is cleared once the QSPI_DATA is read.

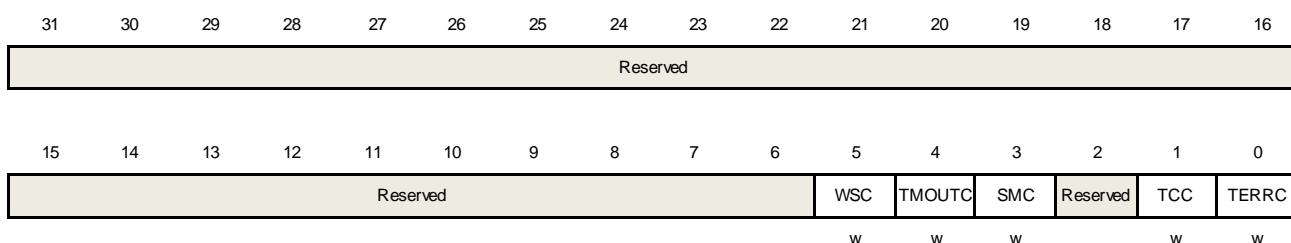
1	TC	Transfer complete flag This bit is set in indirect mode when the programmed number of data has been transmitted or in any mode when abort operation is completed. It is cleared by writing 1 to TCC.
0	TERR	Transfer error flag This bit is set when an invalid address is being accessed in indirect mode. It is cleared by writing 1 to TERRC.

22.11.4. Status clear register (QSPI_STATC)

Address offset: 0x0C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
5	WSC	Clear wrong start sequence flag Writing 1 clears the WS flag in the QSPI_STAT register
4	TMOUTC	Clear timeout flag Writing 1 clears the TMOUT flag in the QSPI_STAT register
3	SMC	Clear status match flag Writing 1 clears the SM flag in the QSPI_STAT register
2	Reserved	Must be kept at reset value
1	TCC	Clear transfer complete flag Writing 1 clears the TC flag in the QSPI_STAT register
0	TERRC	Clear transfer error flag Writing 1 clears the TERR flag in the QSPI_STAT register

22.11.5. Data length register (QSPI_DTLEN)

Address offset: 0x10

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
DTLEN[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTLEN[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DTLEN[31:0]	<p>Data length</p> <p>Number of data to be retrieved (value+1) in indirect and status-polling modes. A value no greater than 3 (indicating 4 bytes) should be used for status-polling mode.</p> <p>All 1s in indirect mode means undefined length, where QSPI will continue until the end of memory, as defined by FMSZ.</p> <ul style="list-style-type: none"> 0x0000_0000: 1 byte is to be transferred 0x0000_0001: 2 bytes are to be transferred 0x0000_0002: 3 bytes are to be transferred 0x0000_0003: 4 bytes are to be transferred ... 0xFFFF_FFFD: 4,294,967,294 (4G-2) bytes are to be transferred 0xFFFF_FFFE: 4,294,967,295 (4G-1) bytes are to be transferred 0xFFFF_FFFF: undefined length -- all bytes until the end of Flash memory (as defined by FMSZ) are to be transferred. Continue reading indefinitely if FMSZ = 0x1F. <p>This field has no effect when in memory-mapped mode.</p> <p>This field can be written only when BUSY = 0.</p>

22.11.6. Transfer configuration register (QSPI_TCFG)

Address offset: 0x14

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		SIOO	FMOD[1:0]	DATAMOD[1:0]	Reserved	DUMYC[4:0]		ALTESZ[1:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEMOD[1:0]		ADDRSZ[1:0]	ADDRMOD[1:0]	IMOD[1:0]	INSTRUCTION[7:0]										

rw	rw	rw	rw	rw
Bits	Fields	Descriptions		
31:29	Reserved	must be kept at reset value		
28	SIOO	Send instruction only once mode This bit has no effect when IMOD = 00. 0: Send instruction on every command sequence 1: Send instruction only for the first command sequence This field can be written only when BUSY = 0.		
27:26	FMOD[1:0]	Functional mode This field defines the QSPI functional mode of operation. 00: Indirect write mode 01: Indirect read mode 10: Status polling mode 11: Memory-mapped mode If DMAEN = 1 already, then the DMA controller for the corresponding channel must be disabled before changing the FMOD value. This field can be written only when BUSY = 0.		
25:24	DATAMOD[1:0]	Data mode This field defines the data phase's mode of operation: 00: No data 01: Data on a single line 10: Data on two lines 11: Data on four lines This field also determines the dummy phase mode of operation. This field can be written only when BUSY = 0.		
23	Reserved	must be kept at reset value		
22:18	DUMYC[4:0]	Number of dummy cycles This field defines the duration of the dummy phase. This field can be written only when BUSY = 0		
17:16	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size: 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes This field can be written only when BUSY = 0.		
15:14	ALTEMOD[1:0]	Alternate bytes mode This field defines the alternate-bytes phase mode of operation: 00: No alternate bytes		

		01: Alternate bytes on a single line 10: Alternate bytes on two lines 11: Alternate bytes on four lines This field can be written only when BUSY = 0.
13:12	ADDRSZ[1:0]	Address size This bit defines address size: 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address This field can be written only when BUSY = 0.
11:10	ADDRMOD[1:0]	Address mode This field defines the address phase mode of operation: 00: No address 01: Address on a single line 10: Address on two lines 11: Address on four lines This field can be written only when BUSY = 0.
9:8	IMOD[1:0]	Instruction mode This field defines the instruction phase mode of operation: 00: No instruction 01: Instruction on a single line 10: Instruction on two lines 11: Instruction on four lines This field can be written only when BUSY = 0.
7:0	INSTRUCTION[7:0]	Instruction Command information to be send to the flash memory. This field can be written only when BUSY = 0.

22.11.7. Address register (QSPI_ADDR)

Address offset: 0x18

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
ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR[31:0]	<p>Address</p> <p>Address to be send to the external Flash memory</p> <p>This bits can only be written when BUSY = 0 and memory-mapped mode is not configured.</p>

22.11.8. Alternate bytes register (QSPI_ALTE)

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
ALTE[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ALTE[31:0]	<p>Alternate Bytes</p> <p>Optional data to be send to the flash memory.</p> <p>This field can be written only when BUSY = 0</p>

22.11.9. Data register (QSPI_DATA)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by word/half word/byte.(32-bits/16-bits/8-bits)

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	<p>Data</p> <p>Data to be transferred through the flash memory.</p> <p>In indirect write mode, data written to this register is stored on the FIFO before sent to the Flash memory. If the FIFO is full, a write operation is stalled until the FIFO</p>

has enough space.

In indirect read mode, reading this register gives the data received from the Flash memory. If the FIFO does not have as many bytes as requested by the read command and if BUSY=1, the read operation is stalled until enough data is present or until the transfer is complete.

In status polling mode, this register contains the last data read from the Flash memory.

22.11.10. Secure Status register (QSPI_STAT_SEC)

Address offset: 0x108

Reset value: 0x0000 0004

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

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
16	DMAEN	<p>DMA enable</p> <p>In indirect mode, DMA can be used to transfer data via QSPI_DATA. DMA transfers are initiated when FT is set.</p> <p>0: DMA disabled</p> <p>1: DMA enabled</p>
11	WSIE	<p>Wrong start sequence interrupt enable</p> <p>This bit enables the wrong start sequence interrupt.</p> <p>0: Interrupt disable</p> <p>1: Interrupt enabled</p>
10	TMOUTIE	<p>Timeout interrupt enable</p> <p>This bit enables the timeout interrupt.</p> <p>0: Interrupt disable</p> <p>1: Interrupt enabled</p>
9	SMIE	<p>Status match interrupt enable</p> <p>This bit enables the status match interrupt.</p>

		0: Interrupt disable 1: Interrupt enabled
8	FTIE	FIFO threshold interrupt enable This bit enables the fifo threshold interrupt. 0: Interrupt disable 1: Interrupt enabled
7	TCIE	Transfer complete interrupt enable This bit enables the transfer complete interrupt. 0: Interrupt disable 1: Interrupt enabled
6	TERRIE	Transfer error interrupt enable This bit enables the transfer error interrupt. 0: Interrupt disable 1: Interrupt enabled
5	WS	Wrong start sequence flag This bit is set when a wrong secure start sequence is detected .This bit is cleared by writing 1 to WSC.
4	TMOUT	Timeout flag This bit is set when timeout occurs. It is cleared by writing 1 to TMOUTC.
3	SM	Status match flag This bit is set in status polling mode when the unmasked received data matches the expected value. It is cleared by writing 1 to SMC.
2	FT	FIFO threshold flag In indirect mode, this bit is set when the FIFO threshold has been reached, or if the FIFO is not empty after the last read operation from the Flash memory. In automatic polling mode this bit is set every time the status register is read from the flash, and it is cleared once the QSPI_DATA is read.
1	TC	Transfer complete flag This bit is set in indirect mode when the programmed number of data has been transmitted or in any mode when abort operation is completed. It is cleared by writing 1 to TCC.
0	TERR	Transfer error flag This bit is set when an invalid address is being accessed in indirect mode. It is cleared by writing 1 to TERRC.

22.11.11. Secure Status clear register (QSPI_STATC_SEC)

Address offset: 0x10C

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								WSC	TMOUTC	SMC	Reserved	TCC	TERRC	W	W

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value
5	WSC	Clear wrong start sequence flag Writing 1 clears the WS flag in the QSPI_STAT register
4	TMOUTC	Clear timeout flag Writing 1 clears the TMOUT flag in the QSPI_STAT register
3	SMC	Clear status match flag Writing 1 clears the SM flag in the QSPI_STAT register
2	Reserved	Must be kept at reset value
1	TCC	Clear transfer complete flag Writing 1 clears the TC flag in the QSPI_STAT register
0	TERRC	Clear transfer error flag Writing 1 clears the TERR flag in the QSPI_STAT register

22.11.12. Secure Data length register (QSPI_DTLEN_SEC)

Address offset: 0x110

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
DTLEN[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTLEN[15:0]															

rw

Bits	Fields	Descriptions
31:0	DTLEN[31:0]	Data length Number of data to be retrieved (value+1) in indirect and status-polling modes. A value no greater than 3 (indicating 4 bytes) should be used for status-polling mode. All 1s in indirect mode means undefined length, where QSPI will continue until the

end of memory, as defined by FMSZ.

0x0000_0000: 1 byte is to be transferred

0x0000_0001: 2 bytes are to be transferred

0x0000_0002: 3 bytes are to be transferred

0x0000_0003: 4 bytes are to be transferred

...

0xFFFF_FFFD: 4,294,967,294 (4G-2) bytes are to be transferred

0xFFFF_FFFE: 4,294,967,295 (4G-1) bytes are to be transferred

0xFFFF_FFFF: undefined length -- all bytes until the end of Flash memory (as defined by FMSZ) are to be transferred. Continue reading indefinitely if FMSZ = 0x1F.

This field has no effect when in memory-mapped mode.

This field can be written only when BUSY = 0.

22.11.13. Secure Transfer configuration register (QSPI_TCFG_SEC)

Address offset: 0x114

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	SIOO	FMOD[1:0]	DATAMOD[1:0]	Reserved	DUMYC[4:0]		ALTESZ[1:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEMOD[1:0]	ADDRSZ[1:0]	ADDRMOD[1:0]	IMOD[1:0]	INSTRUCTION[7:0]											
rw	rw	rw	rw												rw

Bits	Fields	Descriptions
31:29	Reserved	must be kept at reset value
28	SIOO	<p>Send instruction only once mode</p> <p>This bit has no effect when IMOD = 00.</p> <p>0: Send instruction on every command sequence</p> <p>1: Send instruction only for the first command sequence</p> <p>This field can be written only when BUSY = 0.</p>
27:26	FMOD[1:0]	<p>Functional mode</p> <p>This field defines the QSPI functional mode of operation.</p> <p>00: Indirect write mode</p> <p>01: Indirect read mode</p> <p>10: Status polling mode</p> <p>11: Memory-mapped mode</p> <p>If DMAEN = 1 already, then the DMA controller for the corresponding channel must be disabled before changing the FMOD value.</p>

		This field can be written only when BUSY = 0.
25:24	DATAMOD[1:0]	<p>Data mode</p> <p>This field defines the data phase's mode of operation:</p> <ul style="list-style-type: none"> 00: No data 01: Data on a single line 10: Data on two lines 11: Data on four lines <p>This field also determines the dummy phase mode of operation.</p> <p>This field can be written only when BUSY = 0.</p>
23	Reserved	Must be kept at reset value
22:18	DUMYC[4:0]	<p>Number of dummy cycles</p> <p>This field defines the duration of the dummy phase. This field can be written only when BUSY = 0</p>
17:16	ALTESZ[1:0]	<p>Alternate bytes size</p> <p>This bit defines alternate bytes size:</p> <ul style="list-style-type: none"> 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes <p>This field can be written only when BUSY = 0.</p>
15:14	ALTEMOD[1:0]	<p>Alternate bytes mode</p> <p>This field defines the alternate-bytes phase mode of operation:</p> <ul style="list-style-type: none"> 00: No alternate bytes 01: Alternate bytes on a single line 10: Alternate bytes on two lines 11: Alternate bytes on four lines <p>This field can be written only when BUSY = 0.</p>
13:12	ADDRSZ[1:0]	<p>Address size</p> <p>This bit defines address size:</p> <ul style="list-style-type: none"> 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address <p>This field can be written only when BUSY = 0.</p>
11:10	ADDRMOD[1:0]	<p>Address mode</p> <p>This field defines the address phase mode of operation:</p> <ul style="list-style-type: none"> 00: No address 01: Address on a single line 10: Address on two lines 11: Address on four lines

This field can be written only when BUSY = 0.

9:8	IMOD[1:0]	Instruction mode This field defines the instruction phase mode of operation: 00: No instruction 01: Instruction on a single line 10: Instruction on two lines 11: Instruction on four lines This field can be written only when BUSY = 0.
7:0	INSTRUCTION[7:0]	Instruction Command information to be send to the flash memory. This field can be written only when BUSY = 0.

22.11.14. Secure Address register (QSPI_ADDR_SEC)

Address offset: 0x118

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
ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR[31:0]	Address Address to be send to the external Flash memory This bits can only be written when BUSY = 0 and memory-mapped mode is not configured.

22.11.15. Secure Alternate bytes register (QSPI_ALTE_SEC)

Address offset: 0x11C

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
ALTE[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTE[15:0]															

rw

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate Bytes Optional data to be send to the flash memory. This field can be written only when BUSY = 0

22.11.16. Secure Data register (QSPI_DATA_SEC)

Address offset: 0x120

Reset value: 0x0000 0000

This register can be accessed by word/half word/byte.(32-bits/16-bits/8-bits)

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data Data to be transferred through the flash memory. In indirect write mode, data written to this register is stored on the FIFO before sent to the Flash memory. If the FIFO is full, a write operation is stalled until the FIFO has enough space. In indirect read mode, reading this register gives the data received from the Flash memory. If the FIFO does not have as many bytes as requested by the read command and if BUSY=1, the read operation is stalled until enough data is present or until the transfer is complete. In status polling mode, this register contains the last data read from the Flash memory.

22.11.17. Status mask register (QSPI_STATMK)

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
MASK [31:16]															
rw															

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

rw

Bits	Fields	Descriptions
31:0	MASK[31:0]	<p>Status mask</p> <p>Mask to be applied to the status bytes received from the flash memory.</p> <p>For bit n:</p> <ul style="list-style-type: none"> 0: Bit n of the data received is masked and its value is not considered in the matching logic 1: Bit n of the data received is unmasked and its value is considered in the matching logic <p>This field can be written only when BUSY = 0.</p>

22.11.18. Status match register (QSPI_STATMATCH)

Address offset: 0x28

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
MATCH [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH [15:0]															
rw															

Bits	Fields	Descriptions
31:0	MATCH[31:0]	<p>Status match</p> <p>Expected value to be compared with the masked status register to get a match.</p> <p>This field can be written only when BUSY = 0.</p>

22.11.19. Interval register (QSPI_INTERVAL)

Address offset: 0x2C

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	INTERVAL [15:0]	Interval cycle Number of SCK cycles between two read commands in status polling mode. This field can be written only when BUSY = 0.

22.11.20. Timeout register (QSPI_TMOOUT)

Address offset: 0x30

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	TMOOUT[15:0]	Timeout cycle When the FIFO is full in memory mapped mode, this field indicates how many SCK cycles the QSPI waits for next access, keeping CSN low. This field can be written only when BUSY = 0.

22.11.21. FIFO flush register (QSPI_FLUSH)

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

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value

0	FLUSH	Used to flush all qspi interal fifo.
---	-------	--------------------------------------

22.11.22. Wait cnt for indirect wire mode register (QSPI_WTCNT)

Address offset: 0x38

Reset value: 0x0007 A120

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
WTCNT [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WTCNT [15:0]															
rw															

Bits	Fields	Descriptions
31:0	WTCNT[31:0]	Wait cnt when an indirect write operation is completed or aborted. QSPI must keep working after an indirect write operation is completed or aborted until this wait cnt is decreased to zero.

22.11.23. Timeout for status polling mode register (QSPI_SPTMOUT)

Address offset: 0x3C

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
SPTMOUT [[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPTMOUT [15:0]															
rw															

Bits	Fields	Descriptions
31:0	SPTMOUT [31:0]	Timeout cnt when a FMC transfer tries to abort a status polling mode operation. QSPI will not abort status polling mode when a FMC transfer happens. To avoid status polling mode taking too much time, which will result in the blocking of FMC mode operation.

22.11.24. FMC mode security configuration register (QSPI_FMC_SECCFG)

Address offset: 0x7C

Reset value: 0x0000 0000

When the system is secure ($TZEN = 1$), this register provides write access security and can be written only when the access is secure. A non-secure write access is WI and generates an illegal access event. There are no read restrictions. When the system is not secure ($TZEN=0$), this register is RAZ/WI. This register can be protected against non-privileged access when $PRIV=1$ in the **QSPI_PRIVCFG** register.

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															FMCSEC

rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value
0	FMCSEC	FMC mode security When set, the registers of FMC mode(QSPI_CTLF/QSPI_TCFGF/QSPI_ALTEF/QSPI_BYT_CNT) are secure.

22.11.25. Control register in FMC mode (QSPI_CTLF)

Address offset: 0x80

Reset value: 0x8000 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
PSCF[7:0]								CKMODF	Reserved				CSHCF[2:0]		
rw								rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCKDVALUEF[3:0]				Reserved				SCKDENF	SSAMPLEF[1:0]		Reserved				
rw								rw							rw

Bits	Fields	Descriptions
31:24	PSCF [7:0]	This field defines the scaler factor for generating SCK based on the AHB clock in FMC mode (value+1). 0: $F_{CLK} = F_{AHB}$, 1: $F_{CLK} = F_{AHB} / 2$ 2: $F_{CLK} = F_{AHB} 3$... 255: $F_{CLK} = F_{AHB} / 256$ For odd clock division factors, CLK's duty cycle is not 50%. The clock signal remains

		low one cycle longer than it stays high. This field can be modified only when BUSY = 0.
23	CKMODEF	This bit indicates the SCK level in FMC mode when QSPI is free 0: CLK must stay low while CSN is high (QSPI is free). 1: CLK must stay high while CSN is high (QSPI is free). This field can be modified only when BUSY = 0.
22:19	Reserved	Must be kept at reset value
18:16	CSHCF[2:0]	Chip select high cycle in FMC mode CSHC+1 defines the minimum number of CLK cycles which the chip select (CSN) must stay high between two command sequences. 0: CSN stays high for at least 1 cycle between Flash memory commands 1: CSN stays high for at least 2 cycles between Flash memory commands ... 7: CSN stays high for at least 8 cycles between Flash memory commands This field can be modified only when BUSY = 0.
15:12	SCKDVALUEF[3:0]	sck delay value in FMC mode this only useful when SCK_DENF is enable and SSAMPLEF is set
11:7	Reserved	Must be kept at reset value
6	SCKDENF	SCK delay enable when read data from flash in FMC mode, it is only useful when SSAMPLEF is 1 0: SCK delay disabled 1: SCK delay enabled
5:4	SSAMPLEF[1:0]	Sample shift in FMC mode By default, the QSPI samples data 1/2 of a SCK cycle after the data is driven by the Flash memory. This bit allows the data to be sampled later in order to account for external signal delays. 0: No shift 1: 1/2 cycle shift 2: 1 cycle shift 3: Reserved This field can be modified only when BUSY = 0.
3:0	Reserved	Must be kept at reset value

22.11.26. Transfer configuration register in FMC mode (QSPI_TCFGF)

Address offset: 0x84

Reset value: 0x0100 2503

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		SIOOF	Reserved		DATAMODF[1:0]	Reserved	DUMYCF[4:0]			ALTESZF[1:0]						
			rw				rw			rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ALTEMODF[1:0]		ADDRSZF[1:0]	ADDRMODF[1:0]	IMODF[1:0]		INSTRUCTIONF[7:0]			rw							
			rw		rw		rw			rw			rw			

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28	SIOOF	<p>Send instruction only once mode in FMC mode</p> <p>This bit has no effect when IMODE = 0.</p> <p>0: Send instruction on every command sequence</p> <p>1: Send instruction only for the first command sequence</p> <p>This field can be written only when BUSY = 0.</p>
27:26	Reserved	Must be kept at reset value
25:24	DATAMODF[1:0]	<p>Data mode in FMC mode</p> <p>This field defines the data phase's mode of operation:</p> <ul style="list-style-type: none"> 00: No data 01: Data on a single line 10: Data on two lines 11: Data on four lines <p>This field also determines the dummy phase mode of operation.</p> <p>This field can be written only when BUSY = 0.</p>
23	Reserved	Must be kept at reset value
22:18	DUMYCF[4:0]	<p>Number of dummy cycles in FMC mode</p> <p>This field defines the duration of the dummy phase. This field can be written only when BUSY = 0</p>
17:16	ALTESZF[1:0]	<p>Alternate bytes size in FMC mode</p> <p>This bit defines alternate bytes size:</p> <ul style="list-style-type: none"> 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes <p>This field can be written only when BUSY = 0.</p>
15:14	ALTEMODF[1:0]	<p>Alternate bytes mode in FMC mode</p> <p>This field defines the alternate-bytes phase mode of operation:</p> <ul style="list-style-type: none"> 00: No alternate bytes 01: Alternate bytes on a single line 10: Alternate bytes on two lines 11: Alternate bytes on four lines

This field can be written only when BUSY = 0.

13:12	ADDRSZF[1:0]	Address size in FMC mode This bit defines address size: 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address
		This field can be written only when BUSY = 0.
11:10	ADDRMODF[1:0]	Address mode in FMC mode This field defines the address phase mode of operation: 00: No address 01: Address on a single line 10: Address on two lines 11: Address on four lines
		This field can be written only when BUSY = 0.
9:8	IMODF[1:0]	Instruction mode in FMC mode This field defines the instruction phase mode of operation: 00: No instruction 01: Instruction on a single line 10: Instruction on two lines 11: Instruction on four lines
		This field can be written only when BUSY = 0.
7:0	INSTRUCTIONF[7:0]	Instruction in FMC mode Command information to be send to the flash memory. This field can be written only when BUSY = 0.

22.11.27. Alternate bytes register in FMC mode (QSPI_ALTEF)

Address offset: 0x88

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
ALTEF[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEF[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ALTEF[31:0]	Alternate Bytes in FMC mode

Optional data to be send to the flash memory.

This field can be written only when BUSY = 0

22.11.28. Complete bytes counter register (QSPI_BYTE_CNT)

Address offset: 0x8C

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
BYTECNT [31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTECNT [15:0]															
r															

Bits	Fields	Descriptions
31:0	BYTECNT[31:0]	Remained Bytes which has been aborted by FMC mode

22.11.29. Privilege configuration register (QSPI_PRIVCFG)

Address offset: 0x90

Reset value: 0x0000 0000

This register can be read by both privileged and unprivileged access.

When the system is secure (TZEN = 1), this register can be read by secure and non-secure access. It is write-protected against non-secure write access when the bit FMCSEC is set in the QSPI_FMC_SECCFG register. A non-secure write access is ignored and generates an illegal access event.

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

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value
0	PRIV	This bit can be read by both privileged or unprivileged, secure and non-secure access. When set, it can only be cleared by a privileged access. 0: FMC mode registers(QSPI_CTLF/QSPI_TCFGF/QSPI_ALTEF/QSPI_BYTE_CNT) can be

read and written by privileged or unprivileged access.

1: FMC mode registers(QSPI_CTLF/QSPI_TCFGF/QSPI_ALTEF/QSPI_BYTE_CNT) can be read and written by privileged access only.

If the QSPI FMC mode is not secure, the PRIV bit can be written by a secure or non-secure privileged access. If the QSPI FMC mode is secure, the PRIV bit can be written only by a secure privileged access:

- A non-secure write access is ignored and generates an illegal access event.
- A secure unprivileged write access on PRIV bit is ignored.

23. Secure digital input/output interface (SDIO)

23.1. Introduction

The secure digital input/output interface (SDIO) defines the SD/SD I/O /MMC CE-ATA card host interface, which provides command/data transfer between the APB2 system bus and SD memory cards, SD I/O cards, Multimedia Card (MMC), and CE-ATA devices.

The supported SD memory card and SD I/O card system specifications are defined in the SD card Association website at www.sdcard.org.

The supported Multimedia Card system specifications are defined through the Multimedia Card Association website at www.jedec.org, published by the JEDEC SOLID STATE TECHNOLOGY ASSOCIATION.

The supported CE-ATA system specifications are defined through the CE-ATA workgroup website at www.ce-ata.org.

23.2. Main features

The SDIO features include the following:

- **MMC:** Full support for Multimedia Card System Specification Version 4.2(and previous versions) Card and three different data bus modes: 1-bit (default), 4-bit and 8-bit
- **SD Card:** Full support for *SD Memory Card Specifications Version 2.0*
- **SDI/O:** Full support for *SD I/O Card Specification Version 2.0* card and two different data bus modes: 1-bit (default) and 4-bit
- **CE-ATA:** Full compliance with *CE-ATA digital protocol Version 1.1*
- 48MHz data transfer frequency and 8-bit data transfer mode.
- Interrupt and DMA request to processor.
- Completion Signal enables and disable feature (CE-ATA).

Note: SDIO supports only one SD, SD I/O, MMC4.2 card or CE-ATA device at any one time and a stack of MMC4.1 or previous.

23.3. SDIO bus topology

After a power-on reset, the host must initialize the card by a special message-based bus protocol.

Each message is represented by one of the following tokens:

Command: a command is a token which starts an operation. A command is sent from the host to a card. A command is transferred serially on the CMD line.

Response: a response is a token which is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Data: data can be transferred from the card to the host or vice versa. Data is transferred via the data lines. The number of data lines used for the data transfer can be 1(D0), 4(D0-D3) or 8(D0-D7).

The structure of commands, responses and data blocks is described in [Card functional description](#). One data transfer is a bus operation.

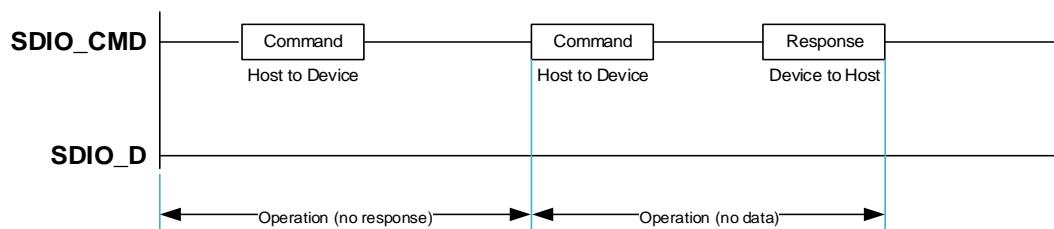
There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have a data token; the others transfer their information directly within the command or response structure. In this case no data token is present in an operation. The bits on the D0-D7 and CMD lines are transferred synchronous to the host clock.

Two types of data transfer commands are defined:

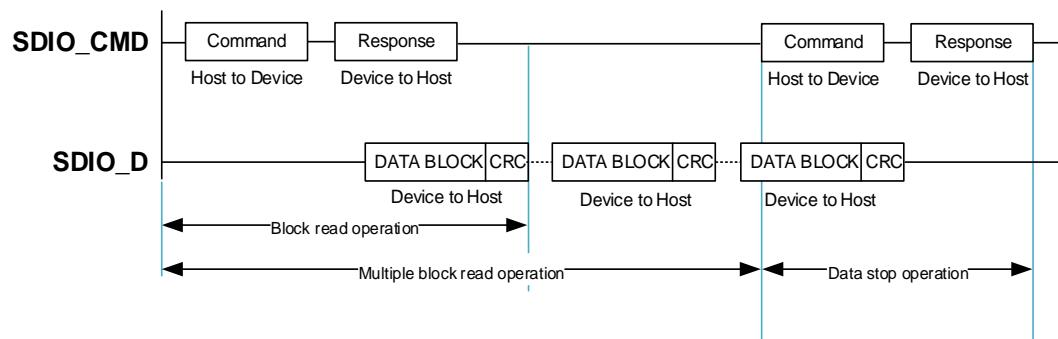
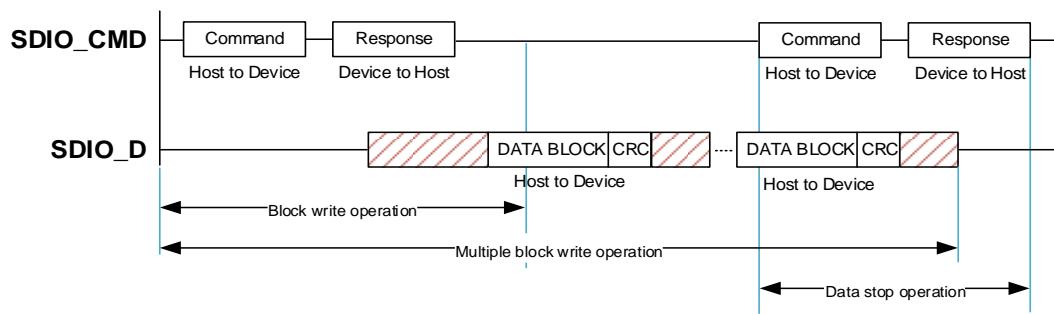
- Stream commands: These commands initiate a continuous data stream; they are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum (only MMC supports).
- Block-oriented commands: These commands send a data block successfully by CRC bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read.

The basic transaction on the bus is the command/response transaction (refer to [Figure 23-1. SDIO “no response” and “no data” operations](#)). This type of bus transaction transfers their information directly within the command or response structure. In addition, some operations have a data token. Data transfers to/from the Card/Device are done in blocks.

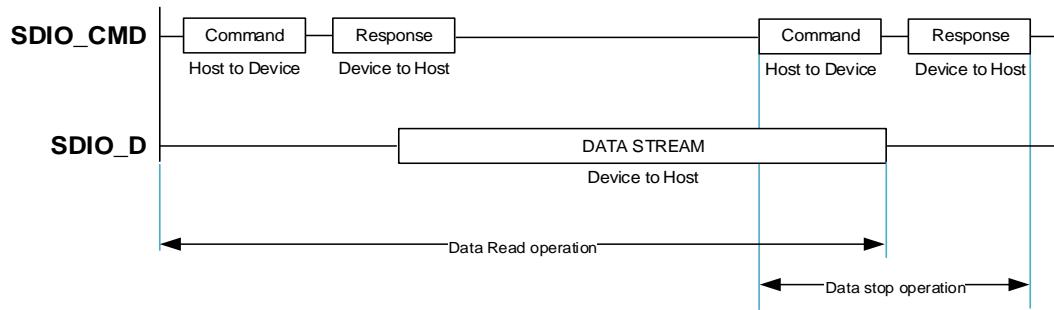
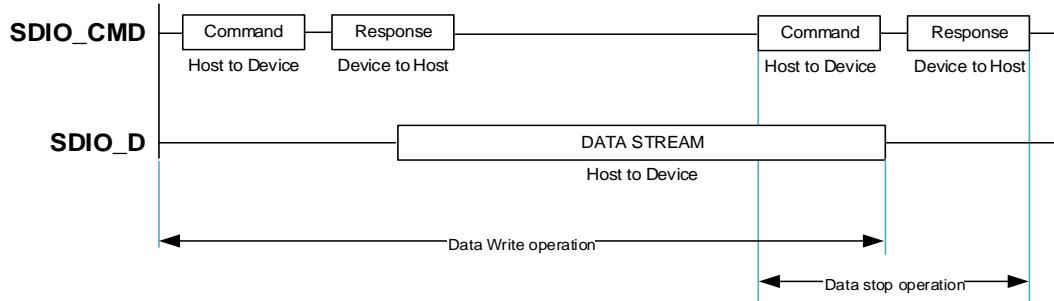
Figure 23-1. SDIO “no response” and “no data” operations



Note that the Multiple Block operation mode is faster than Single Block operation. A multiple block transmission is terminated when a stop command follows on the CMD line. Data transfer can be configured by the host to use single or multiple data lines. [Figure 23-2. SDIO multiple blocks read operation](#) is the multiple blocks read operation and [Figure 23-3. SDIO multiple blocks write operation](#) is the multiple block write operation. The block write operation uses a simple busy signal of the write operation duration on the data (D0) line. CE-ATA device has an optional busy before it is ready to receive the data.

Figure 23-2. SDIO multiple blocks read operation

Figure 23-3. SDIO multiple blocks write operation


Data transfers to/from SD memory cards, SD I/O cards (both IO only card and combo card) and CE-ATA device are done in data blocks. Data transfers to/from MMC are done in data blocks or streams. [Figure 23-4. SDIO sequential read operation](#) and [Figure 23-5. SDIO sequential write operation](#) are the stream read and write operation.

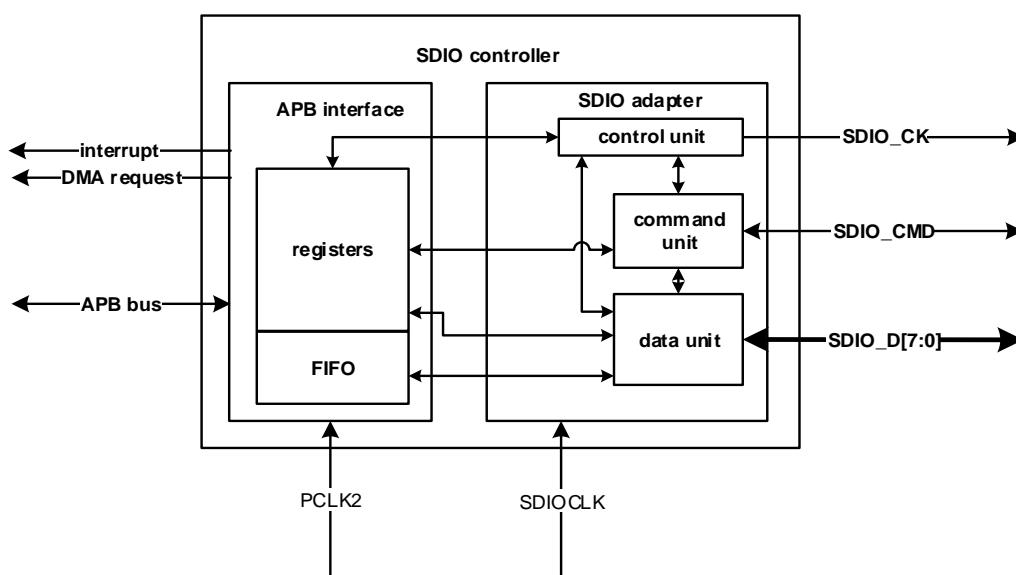
Figure 23-4. SDIO sequential read operation

Figure 23-5. SDIO sequential write operation


23.4. SDIO functional description

The following figure shows the SDIO structure. There have two main parts:

- The SDIO adapter block consists of control unit which manage clock, command unit which manage command transfer, data unit which manage data transfer.
- The APB interface block contains access registers by APB2 bus, contains FIFO unit which is data FIFO used for data transfer, and generates interrupt and DMA request signals.

Figure 23-6. SDIO block diagram



23.4.1. SDIO adapter

The SDIO adapter contains control unit, command unit and data unit, and generates signals to cards. The signals is descript bellow:

SDIO_CK: The SDIO_CK is the clock provided to the card. Each cycle of this signal directs a one bit transfer on the command line (SDIO_CMD) and on all the data lines (SDIO_D). The SDIO_CK frequency can vary between 0 MHz and 20 MHz for a Multimedia Card V3.31, between 0 and 48 MHz for a Multimedia Card V4.2, or between 0 and 25 MHz for an SD/SD I/O card.

The SDIO uses two clock signals: SDIO adapter clock ($\text{SDIOCK} \leq 48\text{MHz}$) and APB2 bus clock (PCLK2)

The frequency of PCLK2 must be no less than the 3/8 frequency of SDIO_CK.

SDIO_CMD: This signal is a bidirectional command channel used for card initialization and transfer of commands. Commands are sent from the SDIO controller to the card and responses are sent from the card to the host. The CMD signal has two operation modes:

open-drain for initialization (only for MMC3.31 or previous), and push-pull for command transfer (SD/SD I/O card MMC4.2 use push-pull drivers also for initialization).

SDIO_D[7:0]: These are bidirectional data channels. The D signals operate in push-pull mode. Only the card or the host is driving these signals at a time. By default, after power up or reset, only D0 is used for data transfer. A wider data bus can be configured for data transfer, using either D0-D3 or D0-D7 (just for MMC4.2), by the SDIO controller. The SDIO includes internal pull-ups for data lines D1-D7. Right after entering to the 4-bit mode the card disconnects the internal pull-ups of lines D1 and D2 (D3 internal pull-up is left connected due to the SPI mode CS usage). Correspondingly right after entering to the 8-bit mode the card disconnects the internal pull-ups of lines D1, D2 and D4-D7.

Table 23-1. SDIO I/O definitions

Pin function	Direction	Description
SDIO_CK	O	SD/SD I/O /MMC clock
SDIO_CMD	I/O	Command input/output
SDIO_D[7:0]	I/O	Data input/output for data lines D[7:0]

The SDIO adapter is an interface to SD/SD I/O /MMC/CE-ATA. It consists of three subunits:

Control unit

The control unit contains the power management functions and the clock management functions for the memory card clock. The power management is controlled by SDIO_PWRCTL register which implements power off or power on. The power saving mode configured by setting CLKPWRSRSAV bit in SDIO_CLKCTL register, which implements close the SDIO_CK when the bus is idle. The clock management generates SDIO_CK to card. The SDIO_CK is generated by a divider of SDIOCLK when CLKBYP bit in SDIO_CLKCTL register is 0, or directly SDIOCLK when CLKBYP bit in SDIO_CLKCTL register is 1.

The Hardware clock control is enabled by setting HWCLKEN in SDIO_CLKCTL register. This functionality is used to avoid FIFO underrun and overrun errors by hardware control the SDIO_CK on/off depending on the system bus is very busy or not. When the FIFO cannot receive or transmit data, the host will stop the SDIO_CK and freeze SDIO state machines to avoid the corresponded error. Only state machines are frozen, the APB2 interface is still alive. So, the FIFO can access by APB2 bus.

Command unit

The command unit implements command transfer to the card. The data transfer flow is controlled by Command State Machine (CSM). After a write operation to SDIO_CMDCTL register and CSMEN in SDIO_CMDCTL register is 1, the command transfer starts. It firstly sends a command to the card. The command contains 48 bits send by SDIO_CMD signal which sends 1 bits to card at one SDIO_CK. The 48 bits command contains 1 bit Start bit, 1 bit Transmission bit, 6 bits command index defined by CMDIDX bits in SDIO_CMDCTL register, 32 bits argument defined in SDIO_CMDAGMT register, 7 bits CRC, and 1 bit end bit.

Then receive response from the card if CMDRESP in SDIO_CMDCTL register is not 0b00/0b10. There are short response which have 48 bits or long response which have 136 bits. The response stores in SDIO_RESP0 - SDIO_RESP3 registers. The command unit also generates the command status flags defined in SDIO_STAT register.

Command state machine

CS_Idle	After reset, ready to send command.	
1.CSM enabled and WAITDEND enabled	→	CS_Pend
2.CSM enabled and WAITDEND disabled	→	CS_Send
3.CSM disabled	→	CS_Idle
Note: The state machine remains in the Idle state for at least eight SDIO_CK periods to meet the Ncc and Nrc timing constraints. Ncc is the minimum delay between two host commands, and Nrc is the minimum delay between the host command and the response.		

CS_Pend	Waits for the end of data transfer.	
1.The data transfer complete	→	CS_Send
2.CSM disabled	→	CS_Idle

CS_Send	Sending the command.	
1.The command transmitted has response	→	CS_Wait
2.The command transmitted doesn't have response	→	CS_Idle
3.CSM disabled	→	CS_Idle

CS_Wait	Wait for the start bit of the response.	
1.Receive the response(detected the start bit)	→	CS_Receive
2.Timeout is reached without receiving the response	→	CS_Idle
3.CSM disabled	→	CS_Idle
Note: The command timeout has a fixed value of 64 SDIO_CK clock periods.		

CS_Receive	Receive the response and check the CRC.	
1.Response Received in CE-ATA mode and interrupt disabled and wait for CE-ATA Command Completion signal enabled	→	CS_Waitcompl
2.Response Received in CE-ATA mode and interrupt disabled and wait for CE-ATA Command Completion signal disabled	→	CS_Pend
3.CSM disabled	→	CS_Idle
4.Response received	→	CS_Idle
5.Command CRC failed	→	CS_Idle

CS_Waitcompl	Wait for the Command Completion signal.		
1.CE-ATA Command Completion signal received	→	CS_Idle	
2.CSM disabled	→	CS_Idle	
3.Command CRC failed	→	CS_Idle	

Data unit

The data unit performs data transfers to and from cards. The data transfer uses SDIO_D[7:0] signals when 8-bits data width (BUSMODE bits in SDIO_CLKCTL register is 0b10), use SDIO_D[3:0] signals when 4-bits data width (BUSMODE bits in SDIO_CLKCTL register is 0b01), or SDIO_D[0] signal when 1-bit data width (BUSMODE bits in SDIO_CLKCTL register is 0b00). The data transfer flow is controlled by Date State Machine (DSM). After a write operation to SDIO_DATACTL register and DATAEN in SDIO_DATACTL register is 1, the data transfer starts. It sends data to card when DATADIR in SDIO_DATACTL register is 0, or receive data from card when DATADIR in SDIO_DATACTL register is 1. The data unit also generates the data status flags defined in SDIO_STAT register.

Data state machine

DS_Idle	The data unit is inactive, waiting for send and receive.		
1.DSM enabled and data transfer direction is from host to card	→	DS_WaitS	
2.DSM enabled and data transfer direction is from card to host	→	DS_WaitR	
3.DSM enabled and Read Wait Started and SD I/O mode enabled	→	DS_Readwait	

DS_WaitS	Wait until the data FIFO empty flag is deasserted or data transfer ended.		
1.Data transfer ended	→	DS_Idle	
2.DSM disabled	→	DS_Idle	
3.Data FIFO empty flag is deasserted	→	DS_Send	

DS_Send	Transmit data to the card.		
1.Data block transmitted	→	DS_Busy	
2.DSM disabled	→	DS_Idle	
3.Data FIFO underrun error occurs	→	DS_Idle	
4. Internal CRC error	→	DS_Idle	

DS_Busy	Waits for the CRC status flag.		
1.Receive a positive CRC status	→	DS_WaitS	
2.Receive a negative CRC status	→	DS_Idle	

3.DSM disabled	→	DS_Idle
4.Timeout occurs	→	DS_Idle
Note: The command timeout programmed in the data timer register (SDIO_DATA TO).		

DS_WaitR	Wait for the start bit of the receive data.	
1.Data receive ended	→	DS_Idle
2.DSM disabled	→	DS_Idle
3.Data timeout reached	→	DS_Idle
4.Receives a start bit before timeout	→	DS_Receive
Note: The command timeout programmed in the data timer register (SDIO_DATA TO).		

DS_Receive	Receive data from the card and write it to the data FIFO.	
1.Data block received	→	DS_WaitR
2.Data transfer ended	→	DS_WaitR
3.Data FIFO overrun error occurs	→	DS_Idle
4.Data received and Read Wait Started and SD I/O mode enabled	→	DS_Readwait
5.DSM disabled or CRC fails	→	DS_Idle

DS_Readwait	Wait for the read wait stop command.	
1.ReadWait stop enabled	→	DS_WaitR
2.DSM disabled	→	DS_Idle

23.4.2. APB2 interface

The APB2 interface implements access to SDIO registers, data FIFO and generates interrupt and DMA request. It includes a data FIFO unit, registers unit, and the interrupt / DMA logic.

The interrupt logic generates interrupt when at least one of the selected status flags is high. An interrupt enable register is provided to allow the logic to generate a corresponding interrupt.

The DMA interface provides a method for fast data transfers between the SDIO data FIFO and memory. The following example describes how to implement this method:

1. Completes the card identification process
2. Increase the SDIO_CK frequency
3. Send CMD7 to select the card and configure the bus width
4. Configure the DMA1 as follows:

Open the DMA1 controller and clear any pending flags. Configure the DMA1_Channel3 or DMA1_Channel6 Peripheral4 source address register with the memory base address and DMA1_Channel3 or DMA1_Channel6 Peripheral4 destination address register with the

SDIO_FIFO register address. Configure DMA1_Channel3 or DMA1_Channel6 Peripheral4 control register (memory with increment transfer, peripheral with not increment transfer, peripheral and memory data size is word size). Program the incremental burst transfer to 4 on peripheral side in DMA1_Channel 3 or DMA1_Channel 6 Peripheral4.

5. Write block to card as follows:

Write the data size in bytes in the SDIO_DATALEN register. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register; the host sends data in blocks of size BLKSZ each. Program SDIO_CMDAGMT register with the data address, where data should be written. Program the SDIO command control register (SDIO_CMDCTL): CMDIDX with 24, CMDRESP with 1 (SDIO card host waits for a short response); CSMEN with '1' (enable to send a command). Other fields are their reset value.

When the CMDRECV flag is set, program the SDIO data control register (SDIO_DATACTL): DATAEN with 1 (enable to send data); DATADIR with 0 (from controller to card); TRANSMOD with 0 (block data transfer); DMAEN with 1 (DMA enabled); BLKSZ with 0x9 (512 bytes). Other bits don't care.

Wait for DTBLKEND flag is set. Check that no channels are still enabled by polling the DMA Interrupt Flag register.

It consists the following subunits:

Register unit

The register unit which contains all system registers generates the signals to control the communication between the controller and card.

Data FIFO

The data FIFO unit has a data buffer, uses as transmit and receive FIFO. The FIFO contains a 32-bit wide, 32-word deep data buffer. The transmit FIFO is used when write data to card and TXRUN in SDIO_STAT register is 1. The data to be transferred is written to transmit FIFO by APB2 bus, the data unit in SDIO adapter read data from transmit FIFO, and then send the data to card. The receive FIFO is used when read data from card and RXRUN in SDIO_STAT register is 1. The data to be transferred is read from the card and then write to receive FIFO. The data in receive FIFO is read to APB2 bus when needed. This unit also generates FIFO flags in SDIO_STAT registers.

23.5. Card functional description

23.5.1. Card registers

Within the card interface registers are defined: OCR, CID, CSD, EXT_CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands. The OCR, CID, CSD and

SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters. The EXT_CSD register carries both, card specific information and actual configuration parameters. For specific information, please refer to the relevant specifications.

OCR register: The 32-bit operation conditions register (OCR) stores the V_{DD} voltage profile of the card and the access mode indication (MMC). In addition, this register includes a status information bit. This status bit is set if the card power up procedure has been finished. The register is a little different between MMC and SD card. The host can use CMD1 (MMC), ACMD41 (SD memory), CMD5 (SD I/O) to get the content of this register.

CID register: The Card Identification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase. Every individual Read/Write (RW) card shall have a unique identification number. The host can use CMD2 and CMD10 to get the content of this register.

CSD register: The Card-Specific Data register provides information regarding access to the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used, etc. The programmable part of the register can be changed by CMD27. The host can use CMD9 to get the content of this register.

Extended CSD Register: Just MMC4.2 has this register. The Extended CSD register defines the card properties and selected modes. It is 512 bytes long. The most significant 320 bytes are the Properties segment, which defines the card capabilities and cannot be modified by the host. The lower 192 bytes are the Modes segment, which defines the configuration the card is working in. These modes can be changed by the host by means of the SWITCH command. The host can use CMD8 (just MMC supports this command) to get the content of this register.

RCA register: The writable 16-bit relative card address register carries the card address that is published by the card during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The host can use CMD3 to ask the card to publish a new relative address (RCA).

Note: The default value of the RCA register is 0x0001(MMC) or 0x0000(SD/SD I/O). The default value is reserved to set all cards into the Stand-by State with CMD7.

DSR register (Optional): The 16-bit driver stage register can be optionally used to improve the bus performance for extended operating conditions (depending on parameters like bus length, transfer rate or number of cards). The CSD register carries the information about the DSR register usage. The default value of the DSR register is 0x404. The host can use CMD4 to get the content of this register.

SCR register: Just SD/SD I/O (if has memory port) have this register. In addition to the CSD register, there is another configuration register named SD CARD Configuration Register (SCR), which is only for SD card. SCR provides information on the SD Memory Card's special features that were configured into the given card. The size of SCR register is 64 bits. This

register shall be set in the factory by the SD Memory Card manufacturer. The host can use ACMD51 to get the content of this register.

23.5.2. Commands

Commands types

There are four kinds of commands defined to control the Card:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr) response from all cards simultaneously
- Addressed (point-to-point) commands (ac) no data transfer on D
- Addressed (point-to-point) data transfer commands (adtc) data transfer on D

Command format

All commands have a fixed code length of 48 bits, as show in [Figure 23-7. Command Token Format](#), needing a transmission time of 1.92 μ s (25 MHz) 0.96 μ s (50 MHz) and 0.92us (52 MHz).

Figure 23-7. Command Token Format

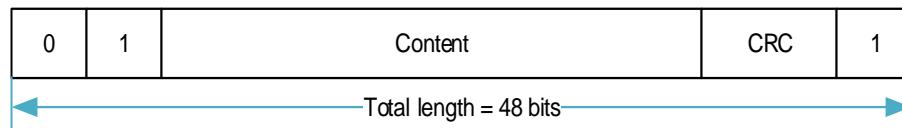


Table 23-2. Command format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

A command always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (host = 1). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by 'x' in the table above indicates this variable is dependent on the command. All commands are protected by a CRC7. Every command code word is terminated by the end bit (always 1).

Command classes

The command set of the Card system is divided into several classes (See [Table 23-3. Card command classes \(CCCs\)](#)). Each class supports a set of card functionalities. [Table 23-3. Card command classes \(CCCs\)](#) determines the setting of CCC from the card supported commands.

For SD cards, Class 0, 2, 4, 5 and 8 are mandatory and shall be supported. Class 7 except

CMD40 is mandatory for SDHC. The other classes are optional. The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

For MMC cards, Class 0 is mandatory and shall be supported. The other classes are either mandatory only for specific card types or optional. By using different classes, several configurations can be chosen (e.g. a block writable card or a stream readable card). The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

For CE-ATA device, the device shall support the MMC commands required to achieve the transfer state during device initialization. Other interface configuration settings, such as bus width, may require additional MMC commands also be supported. See the MMC reference. CE-ATA makes use of the following MMC commands: CMD0 - GO_IDLE_STATE, CMD12 - STOP_TRANSMISSION, CMD39 - FAST_IO, CMD60 - RW_MULTIPLE_REGISTER, CMD61 - RW_MULTIPLE_BLOCK. GO_IDLE_STATE (CMD0), STOP_TRANSMISSION (CMD12), and FAST_IO (CMD39) are as defined in the MMC reference. RW_MULTIPLE_REGISTER (CMD60) and RW_MULTIPLE_BLOCK (CMD61) are MMC commands defined by CE-ATA.

Table 23-3. Card command classes (CCCs)

	Card command class(CCC)	0	1	2	3	4	5	6	7	8	9	10	11
Supported	Class description	basic	Stream read	Block read	Stream write	Block write	erase	write protection	Lock card	application specific	I/O mode	switch	reserved
CMD0	M	+											
CMD1	M	+											
CMD2	M	+											
CMD3	M	+											
CMD4	M	+											
CMD5	O										+		
CMD6	M											+	
CMD7	M	+											
CMD8	M	+											
CMD9	M	+											
CMD10	M	+											
CMD11	M		+										
CMD12	M	+											
CMD13	M	+											

CMD14	M	+											
CMD15	M	+											
CMD16	M			+	+					+			
CMD17	M			+									
CMD18	M			+									
CMD19	M	+											
CMD20	M				+								
CMD23	M			+	+								
CMD24	M				+								
CMD25	M					+							
CMD26	M					+							
CMD27	M					+							
CMD28	M						+						
CMD29	M							+					
CMD30	M							+					
CMD32	M							+					
CMD33	M							+					
CMD34	O											+	
CMD35	O											+	
CMD36	O											+	
CMD37	O											+	
CMD38	M					+							
CMD39												+	
CMD40												+	
CMD42									+				
CMD50	O											+	
CMD52	O											+	
CMD53	O											+	
CMD55	M									+			
CMD56	M									+			
CMD57	O											+	
CMD60	M									+			
CMD61	M									+			
ACMD6	M									+			
ACMD13	M									+			
ACMD22	M									+			
ACMD23	M									+			
ACMD41	M									+			
ACMD42	M									+			
ACMD51	M									+			

Note: 1.CMD1, CMD11, CMD14, CMD19, CMD20, CMD23, CMD26, CMD39 and CMD40 are only available for MMC. CMD5, CMD32-34, CMD50, CMD52, CMD53, CMD57 and ACMDx

are only available for SD card. CMD60, CMD61 are only available for CE-ATA device.

2. All the ACMDs shall be preceded with APP_CMD command (CMD55).

3. CMD8 has different meaning for MMC and SD memory.

Detailed command description

The following tables describe in detail all bus commands. The responses R1-R7 are defined in [Responses](#). The registers CID, CSD and DSR are described in [Card registers](#). The card shall ignore stuff bits and reserved bits in an argument.

Table 23-4. Basic commands (class 0)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all cards to idle state
CMD1	bc	[31:0] OCR without busy	R3	SEND_OP_COND	Asks the card, in idle state, to send its Operating Conditions Register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks any card to send the CID numbers on the CMD line (any card that is connected to the host will respond)
CMD3	bcr	[31:0] stuff bits	R6	SEND_RELATIVE_ADDR	Ask the card to publish a new relative address (RCA)
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards
CMD5	bcr	[31:25]reserved bits [24]S18R [23:0] I/O OCR	R4	IO_SEND_OP_COND	Only for I/O cards. It is similar to the operation of ACMD41 for SD memory cards, used to inquire about the voltage range needed by the I/O card.
CMD6	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Only for MMC. Switches the mode of operation of the selected card or modifies the EXT_CSD registers.

Cmd index	type	argument	Response format	Abbreviation	Description
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/DESELECT_CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnects states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects the card.
CMD8	bcr	[31:12]reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to '0'.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	For MMC only. The card sends its EXT_CSD register as a block of data.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card identification (CID) on CMD the line.
CMD12	ac	[31:0] stuff bits	R1b	STOP TRANSMISSION	Forces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	adtc	[31:0] stuff bits	R1	BUSTEST_R	A host reads the reversed bus testing data pattern from a card.
CMD15	ac	[31:16] RCA [15:0] reserved bits	-	GO_INACTIVE_STATE	Sends an addressed card into the Inactive State. This command is used when the host explicitly wants to deactivate a card.
CMD19	adtc	[31:0] stuff bits	R1	BUSTEST_W	A host sends the bus test data pattern to a card.

Table 23-5. Block-Oriented read commands (class 2)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	<p>In the case of a Standard Capacity SD and MMC, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default is 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD.</p> <p>In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used.</p> <p>In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.</p>
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	<p>In the case of a Standard Capacity SD and MMC, this command reads a block of the size selected by the SET_BLOCKLEN command.</p> <p>In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.</p>
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command. Block length is specified the same as READ_SINGLE_BLOCK command.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 23-6. Stream read commands (class 1) and stream write commands (class 3)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISSION follows.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 23-7. Block-Oriented write commands (class 4)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	See description in Table 23-5. Block-Oriented read commands (class 2) .
CMD23	ac	[31:16] set to 0 [15:0] number of blocks	R1	SET_BLOCK_COUNT	Defines the number of blocks which are going to be transferred in the immediately succeeding multiple block read or write command. If the argument is all 0s, the subsequent read/write operation will be open-ended.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	In the case of a Standard Capacity SD, this command writes a block of the size selected by the SET_BLOCKLEN command. In the case of a SDHC, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.

Cmd index	type	argument	Response format	Abbreviation	Description
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPL E_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows. Block length is specified the same as WRITE_BLOCK command.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
Note: 1. The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD. In the case that write partial blocks is not supported, then the block length=default block length (given in CSD). 2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 23-8. Erase commands (class 5)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK _START	Sets the address of the first write block to be erased.(SD)
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK _END	Sets the address of the last write block of the continuous range to be erased.(SD)
CMD35	ac	[31:0]data address	R1	ERASE_GROUP_ START	Sets the address of the first erase group within a range to be selected for erase.(MMC)
CMD36	ac	[31:0]data address	R1	ERASE_GROUP_ END	Sets the address of the last erase group within a continuous range to be selected for erase.(MMC)
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erases all previously selected write blocks.
Note: 1. CMD34 and CMD37 are reserved in order to maintain backwards compatibility with older					

Cmd index	type	argument	Response format	Abbreviation	Description
versions of the MMC.					
2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 23-9. Block oriented write protection commands (class 6)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). A High Capacity SD Memory Card does not support this command.
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
Note: 1. High Capacity SD Memory Card does not support these three commands.					

Table 23-10. Lock card (class 7)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCK_LEN	See description in Table 23-5. Block-Oriented read commands (class 2) .
CMD42	adtc	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.

Cmd index	type	argument	Response format	Abbreviation	Description
					Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.

Table 23-11. Application-specific commands (class 8)

Cmd index	type	argument	Response format	Abbreviation	Description
ACMD41	bcr	[31]reserved bit [30]HCS [29:24]reserved bits [23:0]V _{DD} Voltage Window (OCR[23:0])	R3	SD_SEND_OP_COND	Sends host capacity support information (HCS) and asks the accessed card to send its operating condition register(OCR) content in the response. HCS is effective when card receives SEND_IF_COND command. CCS bit is assigned to OCR[30].
ACMD42	ac	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/Disconnect[0] the 50K pull-up resistor on CD/D3 (pin 1) of the card.
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits. [0] RD/WR	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose/application specific command. The host sets RD/WR=1 for reading data from the card and sets to 0 for writing data to the card.
CMD60	adtc	[31] WR [23:18] Address [7:2] Byte Count Other bits are	R1(read)/ R1b(write)	RW_MULTIPLE_REGISTER	Read or write register in address range.

Cmd index	type	argument	Response format	Abbreviation	Description
		reserved bits.			
CMD61	adtc	[31] WR [15:0] Data Unit Count Other bits are reserved bits	R1(read)/ R1b(write)	RW_MULTIPLE_BLOCK	Read or write data block in address range.
Note: 1.ACMDx is Application-specific Commands for SD memory. 2. CMD60, CMD61 are Application-specific Commands for CE-ATA device.					

Table 23-12. I/O mode commands (class 9)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD39	ac	[31:16] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8 bit (register) data fields. The command addresses a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the addressed register if the write flag is cleared to 0. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode
CMD52	adtc	[31] R/W Flag [30:28] Function Number [27] RAW Flag [26] Stuff Bits [25:9] Register Address [8] Stuff Bits [7:0] Write Data/Stuff Bits	R5	IO_RW_DIRECT	The IO_RW_DIRECT is the simplest means to access a single register within the total 128K of register space in any I/O function, including the common I/O area (CIA). This command reads or writes 1 byte using only 1 command/response pair. A common use is to initialize registers or monitor status values for I/O functions. This command is the fastest means

Cmd index	type	argument	Response format	Abbreviation	Description
					to read or write single I/O registers, as it requires only a single command/response pair.
CMD53	adtc	[31] R/W Flag [30:28] Function Number [27] Block Mode [26] OP code [25:9] Register Address [8:0] Byte/Block Count		IO_RW_EXTEN DED	This command allows the reading or writing of a large number of I/O registers with a single command.
Note: 1. CMD39, CMD40 are only for MMC. 2. CMD52, CMD53 are only for SD I/O card.					

Table 23-13. Switch function commands (class 10)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD6	adtc	[31] Mode 0:Check function 1:Switch function [30:24] reserved [23:20] reserved for function group 6 (0h or Fh) [19:16] reserved for function group 5 (0h or Fh) [15:12] reserved for function group 4 (0h or Fh) [11:8] reserved for function group 3 (0h or Fh) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Only for SD memory and SD I/O. Checks switchable function (mode 0) and switch card function (mode 1).

23.5.3. Responses

All responses are sent on the CMD line. The response transmission always starts with the left

bit of the bit string corresponding to the response code word. The code length depends on the response type.

Responses types

There are 7 types of responses show as follows.

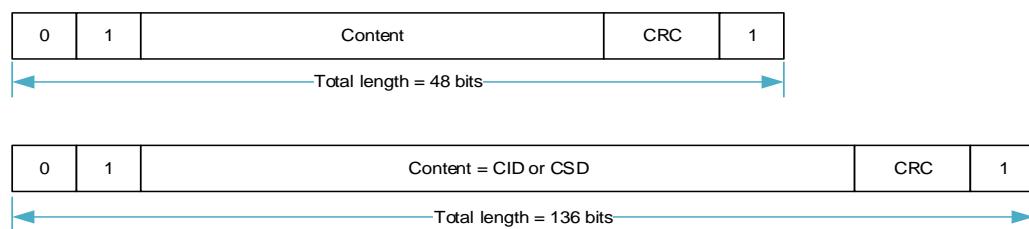
- **R1 / R1b** : normal response command.
- **R2** : CID, CSD register.
- **R3** : OCR register.
- **R4** : Fast I/O.
- **R5** : Interrupt request.
- **R6** : Published RCA response.
- **R7** : Card interface condition.

The SD Memory Card support five types of them, R1 / R1b, R2, R3, R6, R7. And the SD I/O Card and MMC supports additional response types named R4 and R5, but they are not exactly the same for SD I/O Card and MMC.

Responses format

Responses have two formats, as show in [Figure 23-8. Response Token Format](#), all responses are sent on the CMD line. The code length depends on the response type. Except R2 is 136 bits length, others are all 48 bits length.

Figure 23-8. Response Token Format



A response always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (card = 0). A value 'x' in the tables below indicates a variable entry. All responses except for the type R3 are protected by a CRC. Every command code word is terminated by the end bit (always 1).

R1 (normal response command)

Code length is 48 bits. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that if a data transfer to the card is involved, then a busy signal may appear on the data line after the transmission of each block of data. The host shall check for busy after data block transmission. The card status is described in [Data packets format](#).

Table 23-14. Response R1

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	x	x	x	'1'
description	start bit	transmission bit	command index	card status	CRC7	end bit

R1b

R1b is identical to R1 with an optional busy signal transmitted on the data line D0. The card may become busy after receiving these commands based on its state prior to the command reception. The Host shall check for busy at the response.

R2 (CID, CSD register)

Code length is 136 bits. The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127..1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Table 23-15. Response R2

Bit position	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
Value	'0'	'0'	'111111'	x	'1'
description	start bit	transmission bit	reserved	CID or CSD register and internal CRC7	end bit

R3 (OCR register)

Code length is 48 bits. The contents of the OCR register are sent as a response to ACMD41 (SD memory), CMD1 (MMC). The response of different cards may have a little different.

Table 23-16. Response R3

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	'111111'	x	'1111111'	'1'
description	start bit	transmission bit	reserved	OCR register	reserved	end bit

R4 (Fast I/O)

For MMC only. Code length 48 is bits. The argument field contains the RCA of the addressed card, the register address to be read out or written to, and its contents. The status bit in the argument is set if the operation was successful.

Table 23-17. Response R4 for MMC

Bit position	47	46	[45:40]	[39:8] Argument field				[7:1]	0
Width	1	1	6	16	1	7	8	7	1
Value	'0'	'0'	'100111'	x	x	x	x	x	'1'
description	start bit	transmission bit	CMD39	RCA [31:16]	status [15]	register address [14:8]	read register contents [7:0]	CRC7	end bit

R4b

For SD I/O only. Code length is 48 bits. The SDIO card receive the CMD5 will respond with a unique SD I/O response R4.

Table 23-18. Response R4 for SD I/O

Bit position	47	46	[45:40]	39	[38:36]	35	[34:32]	31	[30:8]	[7:1]	0
Width	1	1	6	1	3	1	3	1	23	7	1
Value	'0'	'0'	'1111111'	x	x	x	'000'	x	x	'1111111'	1
description	start bit	transmission bit	Reserved	C	Number of I/O functions	Memory Present	Stuff Bits	S18A	I/O OCR	Reserved	end bit

R5 (Interrupt request)

For MMC only. Code length is 48 bits. If the response is generated by the host, the RCA field in the argument will be 0x0.

Table 23-19. Response R5 for MMC

Bit position	47	46	[45:40]	[39:8] Argument field				[7:1]	0
Width	1	1	6	16	16			7	1
Value	'0'	'0'	'101000'	x	x			x	'1'
description	start bit	transmission bit	CMD40	RCA [31:16] of winning card or of the host	[15:0] Not defined. May be used for IRQ data			CRC7	end bit

R5b

For SD I/O only. The SDIO card's response to CMD52 and CMD53 is R5. If the communication between the card and host is in the 1-bit or 4-bit SD mode, the response shall be in a 48-bit response (R5).

Table 23-20. Response R5 for SD I/O

Bit position	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
Width	1	1	6	16	8	8	7	1

Value	'0'	'0'	'11010X'	'0'	x	x	x	'1'
description	start bit	transmission bit	CMD52/53	Stuff Bits	Response Flags	Read or Write Data	CRC7	end bit

R6 (Published RCA response)

Code length is 48 bit. The bits [45:40] indicate the index of the command to be responded to (CMD3). The 16 MSB bits of the argument field are used for the Published RCA number.

Table 23-21. Response R6

Bit position	47	46	[45:40]	[39:8] Argument field			[7:1]	0
Width	1	1	6	16			7	1
Value	'0'	'0'	'000011'	x			x	'1'
description	start bit	transmission bit	CMD3	New published RCA of the card	card status bits:23,22,19,12:0		CRC7	end bit

R7 (Card interface condition)

For SD memory only. Code length is 48 bits. The card support voltage information is sent by the response of CMD8. Bits 19-16 indicate the voltage range that the card supports. The card that accepted the supplied voltage returns R7 response. In the response, the card echoes back both the voltage range and check pattern set in the argument.

Table 23-22. Response R7

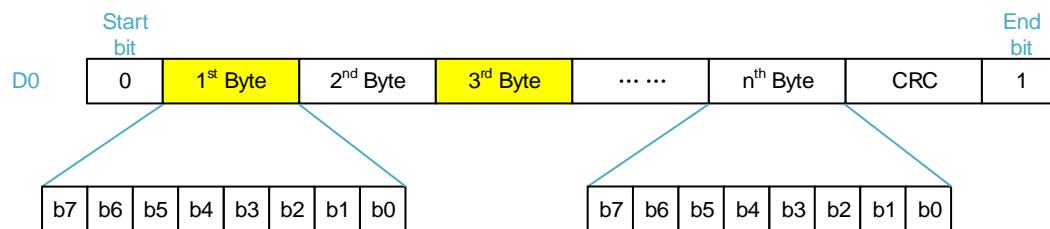
Bit position	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Width	1	1	6	20	4	8	7	1
Value	'0'	'0'	'001000'	'00000h'	x	x	x	'1'
description	start bit	transmission bit	CMD8	Reserved bits	Voltage accepted	echo-back of check pattern	CRC7	end bit

23.5.4. Data packets format

There are 3 data bus mode, 1-bit, 4-bit and 8-bit width. 1-bit mode is mandatory, 4-bit and 8-bit mode is optional. Although using 1-bit mode, D3 also need to notify card current working mode is SDIO or SPI, when card reset and initialize.

1-bit data packet format

After card reset and initialize, only D0 pin is used to transfer data. And other pin can be used freely. [Figure 23-9. 1-bit data bus width](#), [Figure 23-10. 4-bit data bus width](#) and [Figure 23-11. 8-bit data bus width](#) show the data packet format when data bus wide is 1-bit, 4-bit and 8-bit.

Figure 23-9. 1-bit data bus width

4-bit data packet format
Figure 23-10. 4-bit data bus width

	Start bit	1 st Byte		2 nd Byte		3 rd Byte				n th Byte			End bit
D3	0	b7	b3	b7	b3	b7	b3		b7	b3	CRC	1
D2	0	b6	b2	b6	b2	b6	b2		b6	b2	CRC	1
D1	0	b5	b1	b5	b1	b5	b1		b5	b1	CRC	1
D0	0	b4	b0	b4	b0	b4	b0		b4	b0	CRC	1

8-bit data packet format
Figure 23-11. 8-bit data bus width

	Start bit	1 st Byte	2 nd Byte	3 rd Byte						n th Byte			End bit
D7	0	b7	b7	b7					b7	CRC		1
D6	0	b6	b6	b6					b6	CRC		1
D5	0	b5	b5	b5					b5	CRC		1
D4	0	b4	b4	b4					b4	CRC		1
D3	0	b7	b3	b7					b3	CRC		1
D2	0	b6	b2	b6					b2	CRC		1
D1	0	b5	b1	b5					b1	CRC		1
D0	0	b4	b0	b4					b0	CRC		1

23.5.5. Two status fields of the card

The SD Memory supports two status fields and others just support the first one:

Card Status: Error and state information of a executed command, indicated in the response

SD Status: Extended status field of 512 bits that supports special features of the SD Memory Card and future Application-Specific features.

Card status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.

The type and clear condition fields in the table are abbreviated as follows:

Type

- E: Error bit. Send an error condition to the host. These bits are cleared as soon as the response (reporting the error) is sent out.
- S: Status bit. These bits serve as information fields only, and do not alter the execution of the command being responded to. These bits are persistent, they are set and cleared in accordance with the card status.
- R: Exceptions are detected by the card during the command interpretation and validation phase (Response Mode).
- X: Exceptions are detected by the card during command execution phase (Execution Mode).

Clear condition

- A: According to current state of the card.
- B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
- C: Cleared by read

Table 23-23. Card status

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ERX	'0'= no error '1'= error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	ERX	'0'= no error '1'= error	A misaligned address which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	ERX	'0'= no error '1'= error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	ER	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C

Bits	Identifier	Type	Value	Description	Clear Condition
27	ERASE_PARAMETER	ERX	'0'= no error '1'= error	An invalid selection of write-blocks for erase occurred.	C
26	WP_VIOLATION	ERX	'0'= not protected '1'= protected	Set when the host attempts to write to a protected block or to the temporary or permanent write protected card.	C
25	CARD_IS_LOCKED	SX	'0' = card unlocked '1' = card locked	When set, signals that the card is locked by the host	A
24	LOCK_UNLOCK_FAILED	ERX	'0'= no error '1'= error	Set when a sequence or password error has been detected in lock/unlock card command.	C
23	COM_CRC_ERROR	ER	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	ER	'0'= no error '1'= error	Command not legal for the card state.	B
21	CARD_ECC_FAILED	ERX	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	ERX	'0'= no error '1'= error	Internal card controller error.	C
19	ERROR	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	ERX	'0'= no error '1'= error	Only for MMC. The card could not sustain data transfer in stream read mode.	C
17	OVERRUN	ERX	'0'= no error '1'= error	Only for MMC. The card could not sustain data programming in stream write mode.	C
16	CID/ CSD_OVERWRITE	ERX	'0'= no error '1'= error	Can be either one of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP(unprotected)	C

Bits	Identifier	Type	Value	Description	Clear Condition
				bits was made.	
15	WP_ERASE_SKIP	ERX	'0'= not protected '1'= protected	Set when only partial address space was erased due to existing write protected blocks or the temporary or permanent write protected card was erased.	C
14	CARD_ECC_DISABLE	SX	'0'= enabled '1'= disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	SR	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C
[12:9]	CURRENT_STATE	SX	0 = idle 1 = ready 2 = identification 3 = stand by 4 = transfer 5 = send data 6 = receive data 7 = programming 8 = disconnect 9-14 = reserved 15 = reserved for I/O mode	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15.	B
8	READY_FOR_DATA	SX	'0'= not ready '1'= ready	Corresponds to buffer empty signaling on the bus.	A
7	SWITCH_ERROR	EX	'0'= no error '1'= switch error	If set, the card don't switch to the expected mode as requested by the SWITCH command.	B
6	Reserved				
5	APP_CMD	SR	'0'= enabled '1'= disabled	The card will expect ACMD, or an indication that the command has been interpreted as ACMD.	C
4	Reserved				
3	AKE_SEQ_ERROR	ER	'0'= no error '1'= error	Only for SD memory. Error in the sequence of the	C

Bits	Identifier	Type	Value	Description	Clear Condition
				authentication process.	
2	Reserved for application specific commands.				
[1:0]	Reserved for manufacturer test mode.				

Note: 18, 17, 7 bits are only for MMC. 14, 3 bits are only for SD memory.

SD status register

The SD Status contains status bits that are related to the SD Memory Card proprietary features and may be used for future application-specific usage. The size of the SD Status is one data block of 512 bits. The content of this register is transmitted to the Host over the D bus along with a 16-bit CRC. The SD Status is sent to the host over the D bus as a response to ACMD13 (CMD55 followed with CMD13). ACMD13 can be sent to a card only in ‘transfer state’ (card is selected). The SD Status structure is described below.

The same abbreviation for ‘type’ and ‘clear condition’ were used as for the Card Status above.

Table 23-24. SD status

Bits	Identifier	Type	Value	Description	Clear Condition
[511:5 10]	D_BUS_WIDTH	SR	'00'= 1 (default) '01'= reserved '10'= 4 bit width '11'= reserved	Show s the currently defined data bus width that was defined by SET_BUS_WIDTH command	A
509	SECURED_MODE	SR	'0'= Not in the mode '1'= In Secured Mode	Card is in Secured Mode of operation (refer to the “SD Security Specification”).	A
[508:4 96]	reserved				
[495:4 80]	SD_CARD_TYPE	SR	The following cards are currently defined: '0000'= Regular SD RD/WR Card. '0001'= SD ROM '0002'= OTP '0003'= SDHC '0004'= SDXC	In the future, the 8 LSBs will be used to define different variations of an SD Memory Card (Each bit will define different SD Types). The 8 MSBs will be used to define SD Cards that do not comply with current SD Physical Layer Specification.	A
[479:4 48]	SIZE_OF_PROTECT ED_AREA	SR	Size of protected area	(See below)	A
[447:4	SPEED_CLASS	SR	Speed class of the	(See below)	A

Bits	Identifier	Type	Value	Description	Clear Condition
40]			card		
[439:432]	PERFORMANCE_MOVE	SR	Performance of move indicated by 1 [MB/s] step.	(See below)	A
[431:428]	AU_SIZE	SR	Size of AU	(See below)	A
[427:424]			reserved		
[423:408]	ERASE_SIZE	SR	Number of AUs to be erased at a time	(See below)	A
[407:402]	ERASE_TIMEOUT	SR	Timeout value for erasing areas specified by UNIT_OF_ERASE_AU	(See below)	A
[401:400]	ERASE_OFFSET	SR	Fixed offset value added to erase time.	(See below)	A
[399:312]			reserved		
[311:0]			reserved for manufacturer		

SIZE_OF_PROTECTED_AREA

Setting this field differs between SDSC and SDHC/SDXC.

In case of SDSC Card, the capacity of protected area is calculated as follows:

$$\text{Protected Area} = \text{SIZE_OF_PROTECTED_AREA_} * \text{MULT} * \text{BLOCK_LEN}.$$

SIZE_OF_PROTECTED_AREA is specified by the unit in MULT*BLOCK_LEN.

In case of SDHC and SDXC Cards, the capacity of protected area is calculated as follows:

$$\text{Protected Area} = \text{SIZE_OF_PROTECTED_AREA}$$

SIZE_OF_PROTECTED_AREA is specified by the unit in byte.

SPEED_CLASS

This 8-bit field indicates the Speed Class.

00h: Class 0

01h: Class 2

02h: Class 4

03h: Class 6

04h: Class 10

05h–FFh: Reserved

PERFORMANCE_MOVE

This 8-bit field indicates Pm and the value can be set by 1 [MB/sec] step. If the card does not move using RUs, Pm should be considered as infinity. Setting to FFh means infinity. The minimum value of Pm is defined in [Table 23-25. Performance move field](#).

Table 23-25. Performance move field

PERFORMANCE_MOVE	Value Definition
00h	Sequential Write
01h	1 [MB/sec]
02h	2 [MB/sec]
.....
FEh	254 [MB/sec]
FFh	Infinity

AU_SIZE

This 4-bit field indicates AU Size and the value can be selected from 16 KB.

Table 23-26. AU_SIZE field

AU_SIZE	Value Definition
0h	Not Defined
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB
8h	2 MB
9h	4 MB
Ah	8 MB
Bh	12 MB
Ch	16 MB
Dh	24 MB
Eh	32 MB
Fh	64 MB

The maximum AU size, depends on the card capacity, is defined in [Table 23-26. AU_SIZE field](#). The card can set any AU size specified in [Table 23-27. Maximum AU size](#) that is less

than or equal to the maximum AU size. The card should set smaller AU size as possible.

Table 23-27. Maximum AU size

Card Capacity	up to 64MB	up to 256MB	up to 512MB	up to 32GB	up to 2TB
Maximum AU Size	512 KB	1 MB	2 MB	4 MB	64MB

ERASE_SIZE

This 16-bit field indicates N_{ERASE} . When N_{ERASE} of AUs are erased, the timeout value is specified by ERASE_TIMEOUT (Refer to ERASE_TIMEOUT). The host should determine proper number of AUs to be erased in one operation so that the host can indicate progress of erase operation. If this field is set to 0, the erase timeout calculation is not supported.

Table 23-28. Erase size field

ERASE_SIZE	Value Definition
0000h	Erase Time-out Calculation is not supported.
0001h	1 AU
0002h	2 AU
0003h	3 AU
.....
FFFFh	65535 AU

ERASE_TIMEOUT

This 6-bit field indicates the T_{ERASE} and the value indicates erase timeout from offset when multiple AUs are erased as specified by ERASE_SIZE. The range of ERASE_TIMEOUT can be defined as up to 63 seconds and the card manufacturer can choose any combination of ERASE_SIZE and ERASE_TIMEOUT depending on the implementation. Once ERASE_TIMEOUT is determined, it determines the ERASE_SIZE. The host can determine timeout for any number of AU erase by the equation below.

$$\text{Erase timeout of } X \text{ AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad (23-1)$$

Table 23-29. Erase timeout field

ERASE_TIMEOUT	Value Definition
00	Erase Time-out Calculation is not supported.
01	1 [sec]
02	2 [sec]
03	3 [sec]
.....
63	63 [sec]

If ERASE_SIZE field is set to 0, this field shall be set to 0.

ERASE_OFFSET

This 2-bit field indicates the T_{OFFSET} and one of four values can be selected. This field is

meaningless if ERASE_SIZE and ERASE_TIMEOUT fields are set to 0.

Table 23-30. Erase offset field

ERASE_OFFSET	Value Definition
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

23.6. Programming sequence

23.6.1. Card identification

The host will be in card identification mode after reset and while it is looking for new cards on the bus. While in card identification mode the host resets all the cards, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only.

During the card identification process, the card shall operate in the clock frequency of the identification clock rate F_{OD} (400 kHz).

Card reset

The command GO_IDLE_STATE (CMD0) is the software reset command and sets MMC and SD memory card into Idle State regardless of the current card state. The reset command (CMD0) is only used for memory or the memory portion of Combo cards. In order to reset an I/O only card or the I/O portion of a combo card, use CMD52 to write 1 to the RES bit in the CCCR. Cards in Inactive State are not affected by this command.

After power-on by the host, all cards are in Idle State, including the cards that have been in Inactive State before. After power-on or CMD0, all cards' CMD lines are in input mode, waiting for start bit of the next command. The cards are initialized with a default relative card address (RCA) and with a default driver strength with 400 KHz clock frequency.

Operating voltage range validation

At the start of communication between the host and the card, the host may not know the card supported voltage and the card may not know whether it supports the current supplied voltage. To verify the voltage, the following commands are defined in the related specification.

The SEND_OP_COND (CMD1 for MMC), SD_SEND_OP_COND (ACMD41 for SD memory), IO_SEND_OP_COND (CMD5 for SD I/O) command is designed to provide hosts with a mechanism to identify and reject cards which do not match the V_{DD} range desired by the host. This is accomplished by the host sending the required V_{DD} voltage window as the operand of

this command. If the card cannot perform data transfer in the specified range it must discard itself from further bus operations and go into Inactive State. Otherwise, the card shall respond sending back its V_{DD} range.

If the card can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument.

If the card cannot operate on the supplied voltage, it returns no response and stays in idle state. It is mandatory to issue CMD8 prior to ACMD41 to initialize SDHC Card. Receipt of CMD8 makes the cards realize that the host supports the Physical Layer Version 2.00 and the card can enable new functions.

Card identification process

The card identification process differs in different cards. The card can be of the type MMC, CE-ATA, SD, or SD I/O. All types of SD I/O cards are supported, they are, SDIO_IO_ONLY, SDIO_MEM_ONLY, and SDIO COMBO cards. The identification process sequence includes the following steps:

1. Check if the card is connected.
2. Identify the card type; SD, MMC(CE-ATA), or SD I/O.
 - Send CMD5 first. If a response is received, then the card is SD I/O.
 - If not, send ACMD41; if a response is received, then the card is SD.
 - Otherwise, the card is an MMC or CE-ATA.
3. Initialization the card according to the card type.

Use a clock source with a frequency = F_{OD} (that is, 400 KHz) and use the following command sequence:

- SD card - Send CMD0, ACMD41, CMD2, CMD3.
- SDHC card - send CMD0, CMD8, ACMD41, CMD2, CMD3.
- SD I/O - Send CMD52, CMD0, CMD5, if the card doesn't have memory port, send CMD3; otherwise send ACMD41, CMD11 (optional), CMD2, and CMD3.
- MMC/CE-ATA - Send CMD0, CMD1, CMD2, CMD3.

4. Identify the MMC/CE-ATA device.
 - CPU should query the byte 504 (S_CMD_SET) of EXT_CSD register by sending CMD8. If bit 4 is set to 1, then the device supports ATA mode.
 - If ATA mode is supported, the CPU should select the ATA mode by setting the ATA bit (bit 4) in the EXT_CSD register slice 191(CMD_SET) to activate the ATA command set. The CPU selects the command set using the SWITCH (CMD6) command.
 - In the presence of a CE-ATA device, the FAST_IO (CMD39) and RW_MULTIPLE_REGISTER (CMD60) commands will succeed and the returned data will be the CE-ATA reset signature.

23.6.2. No data commands

To send any non-data command, the software needs to program the SDIO_CMDCTL register and the SDIO_CMDAGMT register with appropriate parameters. Using these two registers, the host forms the command and sends it to the command bus. The host reflects the errors in the command response through the error bits of the SDIO_STAT register.

When a response is received the host sets the CMDRECV (CRC check passed) or CCRCERR(CRC check error) bit in the SDIO_STAT register. A short response is copied in SDIO_RESP0, while a long response is copied to all four response registers. The SDIO_RESP3 bit 31 represents the MSB, and the SDIO_RESP0 bit 0 represents the LSB of a long response.

23.6.3. Single block or multiple block write

During block write (CMD24 - 27) one or more blocks of data are transferred from the host to the card. The block consists of start bits(1 or 4 bits LOW), data block, CRC and end bits(1 or 4 bits HIGH). If the CRC fails, the card indicates the failure on the SDIO_D line and the transferred data are discarded and not written, and all further transmitted blocks are ignored.

If the host uses partial blocks whose accumulated length is not block aligned, block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card will detect the block misalignment error before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer. The write operation will also be aborted if the host tries to write data on a write protected area. In this case, however, the card will set the WP_VIOLATION bit (in the status register).

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

Some cards may require long and unpredictable time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the D0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the Disconnect State and release the D line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling D to low if programming is still in progress and the write buffer is unavailable.

For SD card. Setting a number of write blocks to be pre-erased (ACMD23) will make a following Multiple Block Write operation faster compared to the same operation without

preceding ACMD23. The host will use this command to define how many number of write blocks are going to be send in the next write operation.

Steps involved in a single-block or multiple-block write are:

1. Write the data size in bytes in the SDIO_DATALEN register.
2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register; the host sends data in blocks of size BLKSZ.
3. Program SDIO_CMDAGMT register with the data address to which data should be written.
4. Program the SDIO_CMDCTL register. For SD memory and MMC cards, use CMD24 for a single-block write and CMD25 for a multiple-block write. For SD I/O cards, use CMD53 for both single-block and multiple-block transfers. For CE-ATA, first use CMD60 to write the ATA task file, then use CMD61 to write the data. After writing to the CMD register, host starts executing a command, when the command is sent to the bus, the CMDRECV flag is set.
5. Write data to SDIO_FIFO.
6. Software should look for data error interrupts. If required, software can terminate the data transfer by sending the STOP command (CMD12).
7. When a DTEND interrupt is received, the data transfer is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the host should send the STOP command.

23.6.4. Single block or multiple block read

Block read is block oriented data transfer. The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ_BL_LEN), it is always 512 bytes. If READ_BL_PARTIAL(in the CSD) is set, smaller blocks whose starting and ending address are entirely contained within 512 bytes boundary may be transmitted.

CMD17 (READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer state. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. CRC is appended to the end of each block, ensuring data transfer integrity.

Block Length set by CMD16 can be set up to 512 bytes regardless of READ_BL_LEN.

Blocks will be continuously transferred until a STOP_TRANSMISSION command (CMD12) is issued. The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

When the last block of user area is read using CMD18, the host should ignore OUT_OF_RANGE error that may occur even the sequence is correct.

If the host uses partial blocks whose accumulated length is not block aligned and block

misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first misaligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

Steps involved in a single block or multiple block read are:

1. Write the data size in bytes in the SDIO_DATALEN register.
2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register. The host expects data from the card in blocks of size BLKSZ each.
3. Program the SDIO_CMDAGMT register with the data address of the beginning of a data read.
4. Program the SDIO_CMDCTL. For SD and MMC cards, using CMD17 for a single-block read and CMD18 for a multiple-block read. For SD I/O cards, using CMD53 for both single-block and multiple-block transfers. For CE-ATA, first using CMD60 to write the ATA task file, then using CMD 61 to read the data. After writing to the CMD register, the host starts executing the command, when the command is sent to the bus, the CMDRECV flag is set.
5. Software should look for data error interrupts. If required, software can terminate the data transfer by sending a STOP command.
6. The software should read data from the FIFO and make space in the FIFO for receiving more data.
7. When a DTEND interrupt is received, the software should read the remaining data in the FIFO.

23.6.5. Stream write and stream read (MMC only)

Stream write

Stream write (CMD20) starts the data transfer from the host to the card beginning from the starting address until the host issues a stop command. If partial blocks are allowed (if CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. Since the amount of data to be transferred is not determined in advance, CRC cannot be used.

If the host provides an out of range address as an argument to CMD20, the card will reject the command, remain in Tran state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream write command works only on a 1 bit bus configuration (on D0). If CMD20 is issued in other bus configurations, it is regarded as an illegal command.

In order to sustain data transfer in stream mode of the card, the time it takes to receive the data (defined by the bus clock rate) must be less than the time it takes to program it into the main memory field (defined by the card in the CSD register). Therefore, the maximum clock

frequency for the stream write operation is given by the following formula:

$$\text{max write frequency} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}} - 100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (23-2)$$

TRAN_SPEED: Max bus clock frequency.

WRITE_BL_LEN: Max write data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time 1.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the TXURE bit set and return to Transfer state it set and return to Transfer state

Stream read

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its data, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

If the host provides an out of range address as an argument to CMD11, the card will reject the command, remain in Transfer state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream read command works only on a 1 bit bus configuration (on D0). If CMD11 is issued in other bus configurations, it is regarded as an illegal command.

If the end of the memory range is reached while sending data, and no stop command has been sent yet by the host, the contents of the further transferred payload is undefined. As the host sends CMD12 the card will respond with the ADDRESS_OUT_OF_RANGE bit set and return to Tran state.

In order to sustain data transfer in stream mode of the card, the time it takes to transmit the data (defined by the bus clock rate) must be less than the time it takes to read it out of the main memory field (defined by the card in the CSD register). Therefore, the maximum clock frequency for stream read operation is given by the following formula:

$$\text{max read frequency} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}} - 100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (23-3)$$

TRAN_SPEED: Max bus clock frequency.

READ_BL_LEN: Max read data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the RXORE bit set and return to Transfer state bit set and return to Transfer state

23.6.6. Erase

The erasable unit of the MMC/SD memory is the “Erase Group”; Erase group is measured in write blocks which are the basic writable units of the card. The size of the Erase Group is a card specific parameter and defined in the CSD.

The host can erase a contiguous range of Erase Groups. Starting the erase process is a three steps sequence. First the host defines the start address of the range using the ERASE_GROUP_START (CMD35)/ERASE_WR_BLK_START(CMD32) command, next it defines the last address of the range using the ERASE_GROUP_END (CMD36)/ERASE_WR_BLK_END(CMD33) command and finally it starts the erase process by issuing the ERASE (CMD38) command. The address field in the erase commands is an Erase Group address in byte units. The card will ignore all LSB's below the Erase Group size, effectively rounding the address down to the Erase Group boundary.

If an erase command (CMD35, CMD36, and CMD38) is received out of the defined erase sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole sequence.

If the host provides an out of range address as an argument to CMD35 or CMD36, the card will reject the command, respond with the ADDRESS_OUT_OF_RANGE bit set and reset the whole erase sequence.

If an ‘non erase’ command (neither of CMD35, CMD36, CMD38 or CMD13) is received, the card shall respond with the ERASE_RESET bit set, reset the erase sequence and execute the last command.

If the erase range includes write protected blocks, they shall be left intact and only the non-protected blocks shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set.

As described above for block write, the card will indicate that an erase is in progress by holding D0 low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card.

23.6.7. Bus width selection

After the host has verified the functional pins on the bus it should change the bus width configuration.

For MMC, using the SWITCH command (CMD6). The bus width configuration is changed by writing to the BUS_WIDTH byte in the Modes Segment of the EXT_CSD register (using the SWITCH command to do so). After power-on or software reset, the contents of the BUS_WIDTH byte is 0x00. If the host tries to write an invalid value, the BUS_WIDTH byte is not changed and the SWITCH_ERROR bit is set. This register is write only.

For SD memory, using SET_BUS_WIDTH command (ACMD6) to change the bus width. The default bus width after power up or GO_IDLE_STATE command (CMD0) is 1 bit. SET_BUS_WIDTH (ACMD6) is only valid in a transfer state, which means that the bus width can be changed only after a card is selected by SELECT/DESELECT_CARD (CMD7).

23.6.8. Protection management

In order to allow the host to protect data against erase or write, three methods for the cards are supported in the card:

CSD register for card protection (optional)

The entire card may be write protected by setting the permanent or temporary write protect bits in the CSD. Some cards support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD. It is defined in units of WP_GRP_SIZE erase groups as specified in the CSD. The SET_WRITE_PROT command sets the write protection of the addressed write protected group, and the CLR_WRITE_PROT command clears the write protection of the addressed write protected group.

The High Capacity SD Memory Card does not support Write Protection and does not respond to write protection commands (CMD28, CMD29 and CMD30).

Write protect switch on the card (SD memory and SD I/O card)

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is closed the card is not write protected.

Password card Lock/Unlock Operation

The Password Card Lock/Unlock protection is described in [Card Lock/Unlock operation](#).

23.6.9. Card Lock/Unlock operation

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size are kept in a 128-

bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0), ACMD41, CMD16 and lock card command class (class 7). Thus, the host is allowed to reset, initialize, select, query for status, but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0), the card will be locked automatically after power on.

Similar to the existing CSD register write commands, the lock/unlock command is available in "transfer state" only. This means that it does not include an address argument and the card shall be selected before using it.

The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). [Table 23-31. Lock card data structure](#) describes the structure of the command data block.

Table 23-31. Lock card data structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved(all set to 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2								
.....	Password data(PWD)							
PWDS_LEN+1								

ERASE: 1 Defines Forced Erase Operation. In byte 0, bit 3 will be set to 1 (all other bits shall be 0). All other bytes of this command will be ignored by the card.

LOCK/UNLOCK: 1 = Locks the card. 0 = Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).

CLR_PWD: 1 = Clears PWD.

SET_PWD: 1 = Set new password to PWD.

PWDS_LEN: Defines the following password(s) length (in bytes). In case of a password change, this field includes the total password length of old and new passwords. The password length is up to 16 bytes. In case of a password change, the total length of the old password and the new password can be up to 32 bytes.

Password data: In case of setting a new password, it contains the new password. In case of a password change, it contains the old password followed by the new password.

Setting the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the new password. In the case that a password replacement is done, then the block size shall consider that both passwords

(the old and the new one) are sent with the command.

- Send the Card Lock/Unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWDS_LEN) and the password itself. In the case that a password replacement is done, then the length value (PWDS_LEN) shall include both passwords (the old and the new one) and the password data field shall include the old password (currently used) followed by the new password. Note that the card shall handle the calculation of the new password length internally by subtracting the old password length from PWDS_LEN field.
- In the case that the sent old password is not correct (not equal in size and content), then the LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In the case that the sent old password is correct (equal in size and content), then the given new password and its size will be saved in the PWD and PWD_LEN registers, respectively.

Reset the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWDS_LEN), and the password itself. If the PWD and PWD_LEN content match the sent password and its size, then the content of the PWD register is cleared and PWD_LEN is set to 0. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Locking a card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Unlocking the card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

23.7. Specific operations

23.7.1. SD I/O specific operations

The SD I/O only card and SD I/O combo card support these specific operations:

Read Wait operation

Suspend/resume operation

Interrupts

The SD I/O supports these operations only if the SDIO_DATACTL[11] bit is set, except for read suspend that does not need specific hardware implementation.

SD I/O read wait operation

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The Read Wait operation allows a host to signal a card that is executing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SD I/O card. To determine if a card supports the Read Wait protocol, the host shall test SRW capability bit in the Card Capability byte of the CCCR. The timing for Read Wait is based on the Interrupt Period. If a card does not support the Read Wait protocol, the only means a host has to stall (not abort) data in the middle of a read multiple command is to control the SDIO_CK. The limitation of this method is that with the clock stopped, the host cannot issue any commands, and so cannot perform other operations during the delay time. Read Wait support is mandatory for the card to support suspend/resume. [Figure 23-12. Read wait control by stopping SDIO_CK](#) and [Figure 23-13. Read wait operation using SDIO_D\[2\]](#) show the Read Wait mode about stop the SDIO_CK and use SDIO_D[2].

Figure 23-12. Read wait control by stopping SDIO_CK

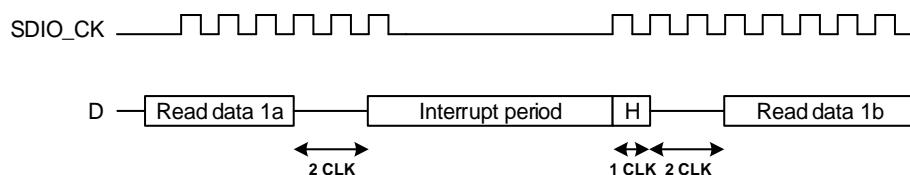
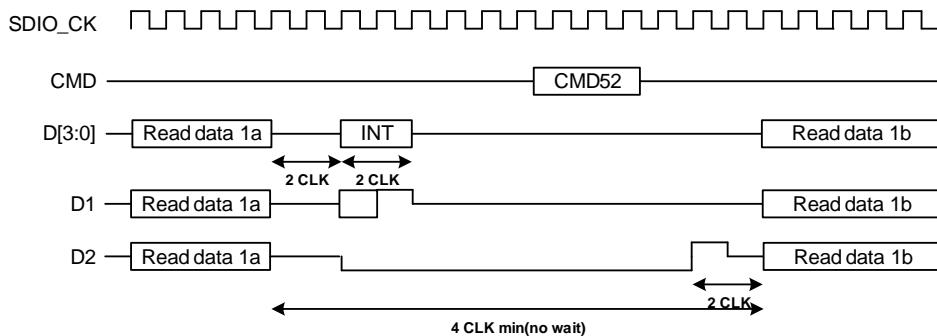


Figure 23-13. Read wait operation using SDIO_D[2]



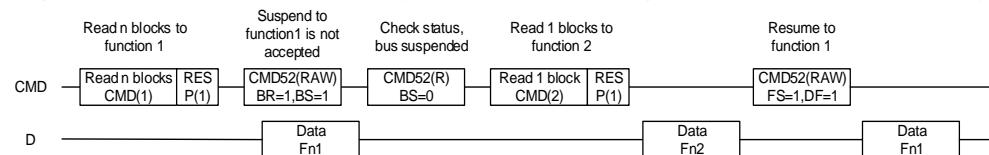
We can start the Read Wait interval before the data block is received: when the data unit is enabled (SDIO_DATACTL[0] bit set), the SD I/O specific operation is enabled (SDIO_DATACTL[11] bit set), Read Wait starts (SDIO_DATACTL[10] = 0 and SDIO_DATACTL[8] = 1) and data direction is from card to SD I/O (SDIO_DATACTL[1] = 1), the DSM directly moves from Idle to Read Wait. In Read Wait the DSM drives SDIO_D[2] to 0 after 2 SDIO_CK clock cycles. In this state, when you set the RWSTOP bit (SDIO_DATACTL[9]), the DSM remains in Wait for two more SDIO_CK clock cycles to drive SDIO_D[2] to 1 for one clock cycle. The DSM then starts waiting again until it receives data from the card. The DSM will not start a Read Wait interval while receiving a block even if Read Wait start is set: the Read Wait interval will start after the CRC is received. The RWSTOP bit has to be cleared to start a new Read Wait operation. During the Read Wait interval, the SDIO can detect SD I/O interrupts on SDIO_D[1].

SD I/O suspend/resume operation

Within a multi-function SD I/O or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SD I/O and combo cards can implement the optional concept of suspend/resume. If a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function or memory. Once this higher-priority transfer is completed, the original transfer is re-started where it left off (resume).

Figure 23-14. Function2 read cycle inserted during function1 multiple read cycle shows a condition where the first suspend request is not immediately accepted. The host then checks the status of the request with a read and determines that the bus has now been released (BS=0). At this time, a read to function 2 is started. Once that single block read is completed, the resume is issued to function 1, causing the data transfer to resume (DF=1).

Figure 23-14. Function2 read cycle inserted during function1 multiple read cycle



When the host sends data to the card, the host can suspend the write operation. The

SDIO_CMDCTL[11] bit is set and indicates to the CSM that the current command is a suspend command. The CSM analyzes the response and when the response is received from the card (suspend accepted), it acknowledges the DSM that goes Idle after receiving the CRC token of the current block.

To suspend a read operation, the DSM waits in the WaitR state, when the function to be suspended sends a complete packet just before stopping the data transaction. The application should continue reading receive FIFO until the FIFO is empty, and the DSM goes Idle state automatically.

Interrupts

In order to allow the SD I/O card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8, which is used as SDIO_D[1] when operating in the 4-bit SD mode, is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SD I/O interrupt is “level sensitive”, that is, the interrupt line shall be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via function unique I/O operation.

When setting the SDIO_DATACTL[11] bit SD I/O interrupts can detect on the SDIO_D[1] line.

Figure 23-15. Read Interrupt cycle timing shows the timing of the interrupt period for single data transaction read cycles.

Figure 23-15. Read Interrupt cycle timing

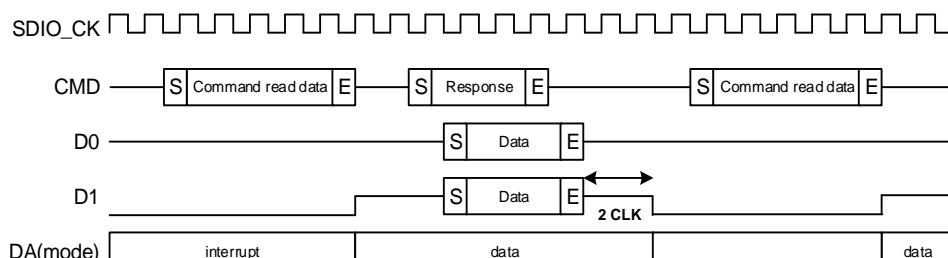
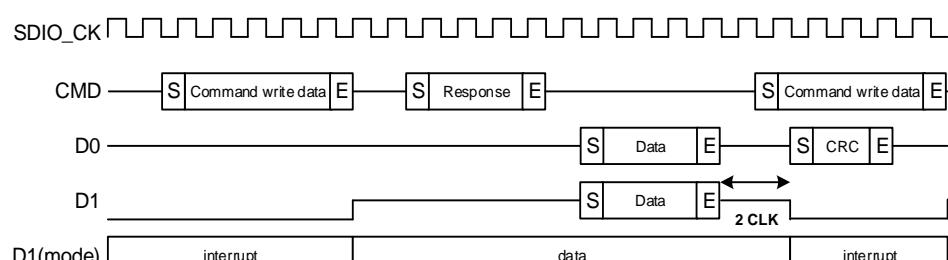


Figure 23-16. Write interrupt cycle timing



When transferring multiple blocks of data in the 4-bit SD mode, a special definition of the interrupt period is required. In order to allow the highest speed of communication, the interrupt period is limited to a 2-clock interrupt period. Card that wants to send an interrupt signal to the host shall assert D1 low for the first clock and high for the second clock. The card shall then release D1 into the hi-Z State. [Figure 23-17. Multiple block 4-Bit read interrupt cycle](#)

[timing](#) shows the operation for an interrupt during a 4-bit multi-block read and [Figure 23-18](#).

[Multiple block 4-Bit write interrupt cycle timing](#) shows the operation for an interrupt during a 4-bit multi-block write

Figure 23-17. Multiple block 4-Bit read interrupt cycle timing

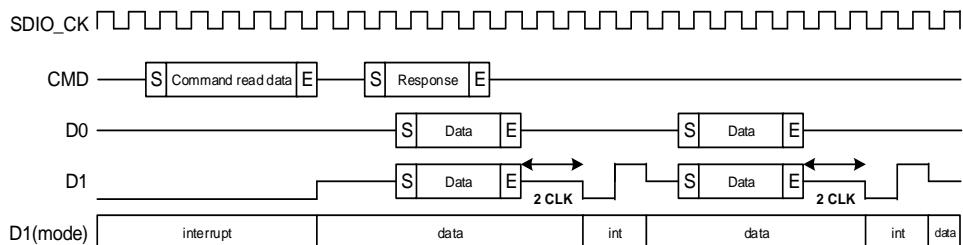
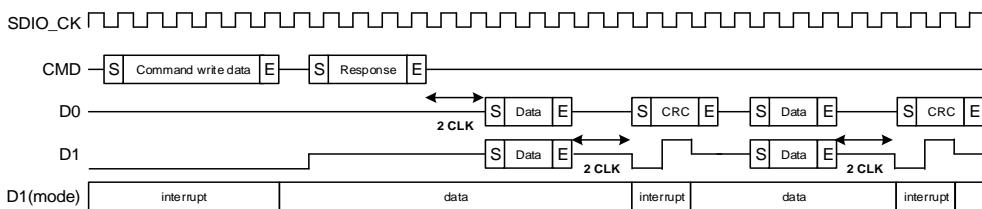


Figure 23-18. Multiple block 4-Bit write interrupt cycle timing



23.7.2. CE-ATA specific operations

The CE-ATA device supports these specific operations:

Receive command completion signal

Send command completion disable signal

The SDIO supports these operations only when SDIO_CMDCTL[14] is set.

Command completion signal

CE-ATA defines a command completion signal that the device uses to notify the host upon normal ATA command completion or when ATA command termination has occurred due to an error condition the device has encountered.

If the ‘enable CMD completion’ bit SDIO_CMDCTL[12] is set and the ‘not interrupt Enable’ bit SDIO_CMDCTL[13] is reset, the CSM waits for the command completion signal in the Waitcompl state.

When start bit is received on the CMD line, the CSM enters the Idle state. No new command can be sent for 7 bit cycles. Then, for the last 5 cycles (out of the 7) the CMD line is driven to‘1’ in push-pull mode.

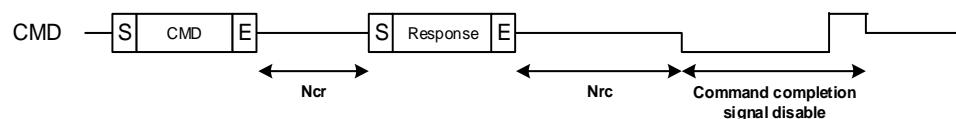
After the host detects a command completion signal from the device, it should issue a FAST_IO (CMD39) command to read the ATA Status register to determine the ending status for the ATA command.

Command completion disable signal

The host may cancel the ability for the device to return a command completion signal by issuing the command completion signal disable. The host shall only issue the command completion signal disable when it has received an R1b response for an outstanding RW_MULTIPLE_BLOCK (CMD61) command.

Command completion signal disable is sent 8 bit cycles after the reception of a short response if the ‘enable CMD completion’ bit, SDIO_CMDCTL[12] is not set and the ‘not interrupt Enable’ bit SDIO_CMDCTL[13] is reset.

Figure 23-19. The operation for command completion disable signal



23.8. SDIO registers

SDIO secure access base address: 0x5001 2C00

SDIO non-secure access base address: 0x4001 2C00

23.8.1. Power control register (SDIO_PWRCTL)

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	PWRCTL[1:0]	SDIO power control bits. These bits control the SDIO state, card input or output. 00: SDIO power off: SDIO cmd/data state machine reset to IDLE, clock to card stopped, no cmd/data output to card 01: Reserved 10: Reserved 11: SDIO Power on

Note: Between two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

23.8.2. Clock control register (SDIO_CLKCTL)

Address offset: 0x04

Reset value: 0x0000 0000

This register controls the output clock SDIO_CK.

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
DIV[8]	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	HWCLKE N	CLKEDG E	BUSMODE[1:0]	CLKBYP	CLKPWR SAV	CLKEN									DIV[7:0]

rw rw rw rw rw rw

Bits	Fields	Descriptions
31	DIV[8]	MSB of Clock division This field defines the MSB division between the input clock (SDIOCLK) and the output clock, refer to bit 7:0 of SDIO_CLKCTL
30:15	Reserved	Must be kept at reset value.
14	HWCLKEN	Hardware Clock Control enable bit If this bit is set, hardware controls the SDIO_CK on/off depending on the system bus is very busy or not. There is no underrun/overrun error when this bit is set, because hardware can close the SDIO_CK when almost underrun/overrun. 0: HW Clock control is disabled 1: HW Clock control is enabled
13	CLKEDGE	SDIO_CK clock edge selection bit 0: Select the rising edge of the SDIOCLK to generate SDIO_CK 1: Select the falling edge of the SDIOCLK to generate SDIO_CK
12:11	BUSMODE[1:0]	SDIO card bus mode control bit 00: 1-bit SDIO card bus mode selected 01: 4-bit SDIO card bus mode selected 10: 8-bit SDIO card bus mode selected
10	CLKBYP	Clock bypass enable bit This bit defines the SDIO_CK is directly SDIOCLK or not. 0: NO bypass, the SDIO_CK refers to DIV bits in SDIO_CLKCTL register. 1: Clock bypass, the SDIO_CK is directly from SDIOCLK (SDIOCLK/1).
9	CLKPWRSAV	SDIO_CK clock dynamic switch on/off for power saving. This bit controls SDIO_CK clock dynamic switch on/off when the bus is idle for power saving 0: SDIO_CK clock is always on 1: SDIO_CK closed when bus idle
8	CLKEN	SDIO_CK clock output enable bit 0: SDIO_CK is disabled 1: SDIO_CK is enabled
7:0	DIV[7:0]	Clock division This field and DIV[8] bit defines the division factor to generate SDIO_CK clock to card. The SDIO_CK is divider from SDIOCLK if CLKBYP bit is 0, and the SDIO_CK frequency = SDIOCLK / (DIV[8:0] + 2).

Note: Between two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

23.8.3. Command argument register (SDIO_CMDAGMT)

Address offset: 0x08

Reset value: 0x0000 0000

This register defines 32 bit command argument, which will be used as part of the command (bit 39 to bit 8).

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
CMDAGMT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDAGMT[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CMDAGMT[31:0]	<p>SDIO card command argument</p> <p>This field defines the SDIO card command argument which sent to card. This field is the bits [39:8] of command message. If the command message contains an argument, this field must update before writing SDIO_CMDCTL register when sending a command.</p>

23.8.4. Command control register (SDIO_CMDCTL)

Address offset: 0x0C

Reset value: 0x0000 0000

The SDIO_CMDCTL register contains the command index and other command control bits to control command state machine.

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

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	ATAEN	<p>CE-ATA command enable(CE-ATA only)</p> <p>If this bit is set, the host enters the CE-ATA mode, and the CSM transfers CMD61.</p> <p>0: CE-ATA disable</p>

		1: CE-ATA enable
13	NINTEN	<p>No CE-ATA Interrupt (CE-ATA only)</p> <p>This bit defines if there is CE-ATA interrupt or not. This bit is only used when CE-ATA card.</p>
		0: CE-ATA interrupt enable
		1: CE_ATA interrupt disable
12	ENCMDC	<p>CMD completion signal enabled (CE-ATA only)</p> <p>This bit defines if there is command completion signal or not in CE-ATA card.</p>
		0: no completion signal
		1: have completion signal
11	SUSPEND	<p>SD I/O suspend command(SD I/O only)</p> <p>This bit defines whether the CSM to send a suspend command or not. This bit is only used for SDIO card.</p>
		0: no effect
		1: suspend command
10	CSMEN	<p>Command state machine (CSM) enable bit</p>
		0: Command state machine disable (stay on CS_Idle)
		1: Command state machine enable
9	WAITDEND	<p>Waits for ends of data transfer.</p> <p>If this bit is set, the command state machine starts to send a command must wait the end of data transfer.</p>
		0: no effect
		1: Wait the end of data transfer
8	INTWAIT	<p>Interrupt wait instead of timeout</p> <p>This bit defines the command state machine to wait card interrupt at CS_Wait state in command state machine. If this bit is set, no command wait timeout generated.</p>
		0: Not wait interrupt.
		1: Wait interrupt.
7:6	CMDRESP[1:0]	<p>Command response type bits</p> <p>These bits define the response type after sending a command message.</p>
		00: No response
		01: Short response
		10: No response
		11: Long response
5:0	CMDIDX[5:0]	<p>Command index</p> <p>This field defines the command index to be sent to SDIO card.</p>

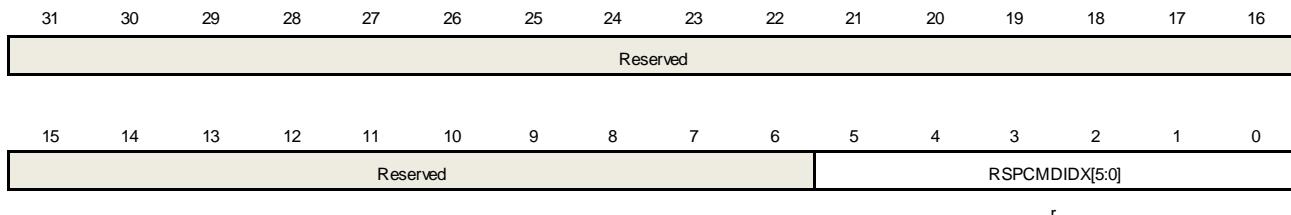
Note: Between Two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

23.8.5. Command index response register (SDIO_RSPCMDIDX)

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.
5:0	RSPCMDIDX[5:0]	Last response command index Read-only bits field. This field contains the command index of the last command response received. If the response doesn't have the command index (long response and short response of R3), the content of this register is undefined.

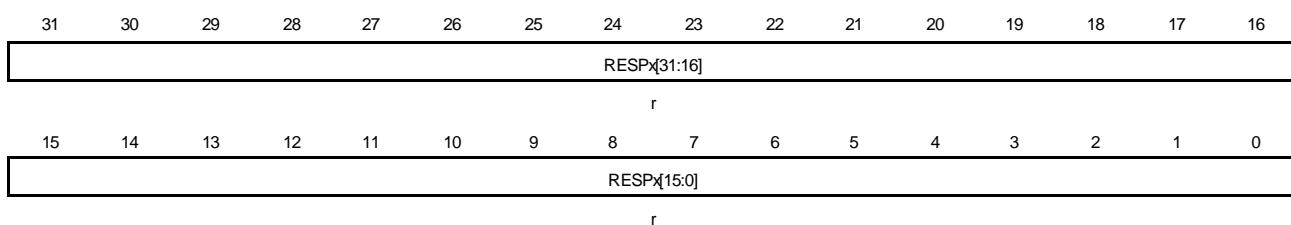
23.8.6. Response register (SDIO_RESPx x=0..3)

Address offset: 0x14+(4*x), x=0..3

Reset value: 0x0000 0000

These register contains the content of the last card response received.

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



Bits	Fields	Descriptions
31:0	RESPx[31:0]	Card state. The content of the response, see Table 23-32. SDIO_RESPx register at different response type .

The short response is 32 bits, the long response is 127 bits (bit 128 is the end bit 0).

Table 23-32. SDIO_RESPx register at different response type

Register	Short response	Long response
SDIO_RESP0	Card response[31:0]	Card response[127:96]

Register	Short response	Long response
SDIO_RESP1	reserved	Card response [95:64]
SDIO_RESP2	reserved	Card response [63:32]
SDIO_RESP3	reserved	Card response [31:1],plus bit 0

23.8.7. Data timeout register (SDIO_DATATO)

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
DATATO[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATATO[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DATATO[31:0]	<p>Data timeout period</p> <p>These bits define the data timeout period count by SDIO_CK. When the DSM enter the state WaitR or BUSY, the internal counter which loads from this register starts decrement. The DSM timeout and enter the state Idle and set the DTTMOUT flag when the counter decreases to 0.</p>

Note: The data timer register and the data length register must be updated before being written to the data control register when need a data transfer.

23.8.8. Data length register (SDIO_DATALEN)

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								DATALEN[24:16]							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATALEN[15:0]															
rw															

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.

24:0	DATALEN[24:0]	Data transfer length This register defined the number of bytes to be transferred. When the data transfer starts, the data counter loads this register and starts decrement.
------	----------------------	--

Note: If block data transfer selected, the content of this register must be a multiple of the block size (refer to SDIO_DA TACTL). The data timer register and the data length register must be updated before being written to the data control register when need a data transfer.

23.8.9. Data control register (SDIO_DATACTL)

Address offset: 0x2C

Reset value: 0x0000 0000

This register controls the DSM.

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		IOEN	RWTYP	RWSTOP	RWEN		BLKSZ[3:0]		DMAEN	TRANSMOD	DATADIR	DATAEN			

rw rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	IOEN	SD I/O specific function enable(SD I/O only) 0: Not SD I/O specific function 1: SD I/O specific function
10	RWTYP	Read wait type(SD I/O only) 0: Read Wait control using SDIO_D[2] 1: Read Wait control by stopping SDIO_CK
9	RWSTOP	Read wait stop(SD I/O only) 0: No effect 1: Stop the read wait process if RWEN bit is set
8	RWEN	Read wait mode enabled(SD I/O only) 0: Read wait mode is disabled 1: Read wait mode is enabled
7:4	BLKSZ[3:0]	Data block size These bits defined the block size when data transfer is block transfer. 0000: block size = 2^0 = 1 byte 0001: block size = 2^1 = 2 bytes 0010: block size = 2^2 = 4 bytes

0011: block size = 2^3 = 8 bytes
 0100: block size = 2^4 = 16 bytes
 0101: block size = 2^5 = 32 bytes
 0110: block size = 2^6 = 64 bytes
 0111: block size = 2^7 = 128 bytes
 1000: block size = 2^8 = 256 bytes
 1001: block size = 2^9 = 512 bytes
 1010: block size = 2^{10} = 1024 bytes
 1011: block size = 2^{11} = 2048 bytes
 1100: block size = 2^{12} = 4096 bytes
 1101: block size = 2^{13} = 8192 bytes
 1110: block size = 2^{14} = 16384 bytes
 1111: reserved

3	DMAEN	DMA enable bit 0: DMA is disabled. 1: DMA is enabled.
2	TRANSMOD	Data transfer mode 0: Block transfer 1: Stream transfer or SDIO multibyte transfer
1	DATADIR	Data transfer direction 0: Write data to card. 1: Read data from card.
0	DATAEN	Data transfer enable bit Write 1 to this bit to start data transfer regardless this bit is 0 or 1. The DSM moves to Readwait state if RWEN is set or to the WaitS, WaitR state depend on DATA DIR bit. Start a new data transfer, it not need to clear this bit to 0.

Note: Between Two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

23.8.10. Data counter register (SDIO_DATACNT)

Address offset: 0x30

Reset value: 0x0000 0000

This register is read only. When the DSM from Idle to WaitR or WaitS, it loads value from data length register (SDIO_DATALEN). It decrements with the data transferred, when it reaches 0, the flag DTEND is set.

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								DATACNT[24:16]							

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

r

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	DATACNT[24:0]	Data count value Read-only bits field. When these bits are read, the number of remaining data bytes to be transferred is returned.

23.8.11. Status register (SDIO_STAT)

Address offset: 0x34

Reset value: 0x0000 0000

This register is read only. The following descripts the types of flag:

The flags of bit [23:22, 10:0] can only be cleared by writing 1 to the corresponding bit in interrupt clear register (SDIO_INTC).

The flags of bit [21:11] are changing depend on the hardware logic.

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								ATAEND	SDIOINT	RXDTVAL	TXDTVAL	RFE	TFE	RFF	TFF
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFH	TFH	RXRUN	TXRUN	CMDRUN	DTBLKE ND	STBITE	DTEND	CMDSEN D	CMDREC V	RXORE	TXURE	DTTMOU T	CMDTMO UT	DTCRCE RR	CCRCER R
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	ATAEND	CE-ATA command completion signal received (only for CMD61)
22	SDIOINT	SD I/O interrupt received
21	RXDTVAL	Data is valid in receive FIFO
20	TXDTVAL	Data is valid in transmit FIFO
19	RFE	Receive FIFO is empty
18	TFE	Transmit FIFO is empty When HW Flow control is enabled, TFE signals becomes activated when the FIFO contains 2 words.
17	RFF	Receive FIFO is full

When HW Flow control is enabled, RFF signals becomes activated 2 words before the FIFO is full.

16	TFF	Transmit FIFO is full
15	RFH	Receive FIFO is half full: at least 8 words can be read in the FIFO
14	TFH	Transmit FIFO is half empty: at least 8 words can be written into the FIFO
13	RXRUN	Data reception in progress
12	TXRUN	Data transmission in progress
11	CMDRUN	Command transmission in progress
10	DTBLKEND	Data block sent/received (CRC check passed)
9	STBITE	Start bit error in the bus.
8	DTEND	Data end (data counter, SDIO_DATACNT, is zero)
7	CMDSEND	Command sent (no response required)
6	CMDRECV	Command response received (CRC check passed)
5	RXORE	Received FIFO overrun error occurs
4	TXURE	Transmit FIFO underrun error occurs
3	DTTMOUT	Data timeout The data timeout period depends on the SDIO_DATATO register.
2	CMDTMOUT	Command response timeout The command timeout period has a fixed value of 64 SDIO_CK clock periods.
1	DTCRCERR	Data block sent/received (CRC check failed)
0	CCRCERR	Command response received (CRC check failed)

23.8.12. Interrupt clear register (SDIO_INTC)

Address offset: 0x38

Reset value: 0x0000 0000

This register is write only. Writing 1 to the bit can clear the corresponding bit in the SDIO_STAT register.

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

Reserved	DTBLKE NDC	STBITEC	DTENDC	CMDSEN DC	CMDREC VC	RXOREC	TXUREC	DTTMOU TC	CMDTMO UTC	DTCRCE	CCRCER RC
	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	ATAENDC	ATAEND flag clear bit Write 1 to this bit to clear the flag.
22	SDIOINTC	SDIOINT flag clear bit Write 1 to this bit to clear the flag.
21:11	Reserved	Must be kept at reset value.
10	DTBLKENDC	DTBLKEND flag clear bit Write 1 to this bit to clear the flag.
9	STBITEC	STBITE flag clear bit Write 1 to this bit to clear the flag.
8	DTENDC	DTEND flag clear bit Write 1 to this bit to clear the flag.
7	CMDSENDC	CMDSEND flag clear bit Write 1 to this bit to clear the flag.
6	CMDRECV C	CMDRECV flag clear bit Write 1 to this bit to clear the flag.
5	RXOREC	RXORE flag clear bit Write 1 to this bit to clear the flag.
4	TXUREC	TXURE flag clear bit Write 1 to this bit to clear the flag.
3	DTTMOUTC	DTTMOUT flag clear bit Write 1 to this bit to clear the flag.
2	CMDTMOUTC	CMDTMOU flag clear bit Write 1 to this bit to clear the flag.
1	DTCRCERRC	DTCRCERR flag clear bit Write 1 to this bit to clear the flag.
0	CCRCERRC	CCRCERR flag clear bit Write 1 to this bit to clear the flag.

23.8.13. Interrupt enable register (SDIO_INTEN)

Address offset: 0x3C

Reset value: 0x0000 0000

This register enables the corresponding interrupt in the SDIO_STAT register.

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								ATAENDIE	SDIOINTIE	RXDVTVAIE	TXDTVALIE	RFEIE	TFEIE	RFFIE	TFFIE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFHIE	TFHIE	RXRUNIE	TXRUNIE	CMDRUNIE	DTBLKE NDIE	STBITEIE	DTENDIE	CMDSEN DIE	CMDREC VIE	RXOREIE	TXUREIE	DTTMOU TIE	CMDTMO UTIE	DTCRCE RRIE	CCRCER RIE
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	ATAENDIE	CE-ATA command completion signal received interrupt enable Write 1 to this bit to enable the interrupt.
22	SDIOINTIE	SD I/O interrupt received interrupt enable Write 1 to this bit to enable the interrupt.
21	RXDVTVALIE	Data valid in receive FIFO interrupt enable Write 1 to this bit to enable the interrupt.
20	TXDTVALIE	Data valid in transmit FIFO interrupt enable Write 1 to this bit to enable the interrupt.
19	RFEIE	Receive FIFO empty interrupt enable Write 1 to this bit to enable the interrupt.
18	TFEIE	Transmit FIFO empty interrupt enable Write 1 to this bit to enable the interrupt.
17	RFFIE	Receive FIFO full interrupt enable Write 1 to this bit to enable the interrupt.
16	TFFIE	Transmit FIFO full interrupt enable Write 1 to this bit to enable the interrupt.
15	RFHIE	Receive FIFO half full interrupt enable Write 1 to this bit to enable the interrupt.
14	TFHIE	Transmit FIFO half empty interrupt enable Write 1 to this bit to enable the interrupt.
13	RXRUNIE	Data reception interrupt enable

		Write 1 to this bit to enable the interrupt.
12	TXRUNIE	Data transmission interrupt enable Write 1 to this bit to enable the interrupt.
11	CMDRUNIE	Command transmission interrupt enable Write 1 to this bit to enable the interrupt.
10	DTBLKENDIE	Data block end interrupt enable Write 1 to this bit to enable the interrupt.
9	STBITEIE	Start bit error interrupt enable Write 1 to this bit to enable the interrupt.
8	DTENDIE	Data end interrupt enable Write 1 to this bit to enable the interrupt.
7	CMDSENDIE	Command sent interrupt enable Write 1 to this bit to enable the interrupt.
6	CMDRECVIE	Command response received interrupt enable Write 1 to this bit to enable the interrupt.
5	RXOREIE	Received FIFO overrun error interrupt enable Write 1 to this bit to enable the interrupt.
4	TXUREIE	Transmit FIFO underrun error interrupt enable Write 1 to this bit to enable the interrupt.
3	DTTMOUTIE	Data timeout interrupt enable Write 1 to this bit to enable the interrupt.
2	CMDTMOUTIE	Command response timeout interrupt enable Write 1 to this bit to enable the interrupt.
1	DTCRCERRIE	Data CRC fail interrupt enable Write 1 to this bit to enable the interrupt.
0	CCRCERRIE	Command response CRC fail interrupt enable Write 1 to this bit to enable the interrupt.

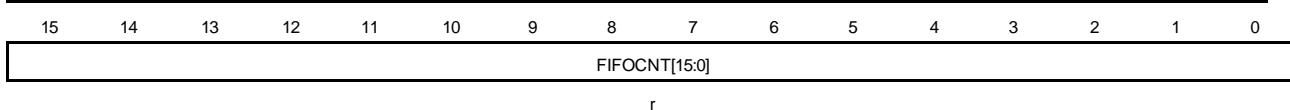
23.8.14. FIFO counter register (SDIO_FIFOCNT)

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
Reserved								FIFOCNT[23:16]							



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	FIFOCNT[23:0]	FIFO counter. These bits define the remaining number words to be written or read from the FIFO. It loads the data length register (SDIO_DATALEN[24:2] if SDIO_DATALEN is word-aligned or SDIO_DATALEN[24:2]+1 if SDIO_DATALEN is not word-aligned) when DATAEN is set, and start count decrement when a word write to or read from the FIFO.

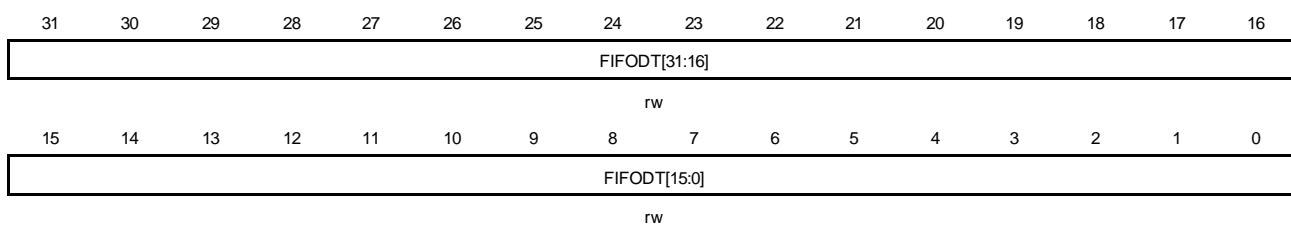
23.8.15. FIFO data register (SDIO_FIFO)

Address offset: 0x80

Reset value: 0x0000 0000

This register occupies 32 entries of 32-bit words, the address offset is from 0x80 to 0xFC.

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



Bits	Fields	Descriptions
31:0	FIFODT[31:0]	Receive FIFO data or transmit FIFO data These bits are the data of receive FIFO or transmit FIFO. Write to or read from this register is write data to FIFO or read data from FIFO.

24. Universal serial bus full-speed interface (USBFS)

24.1. Overview

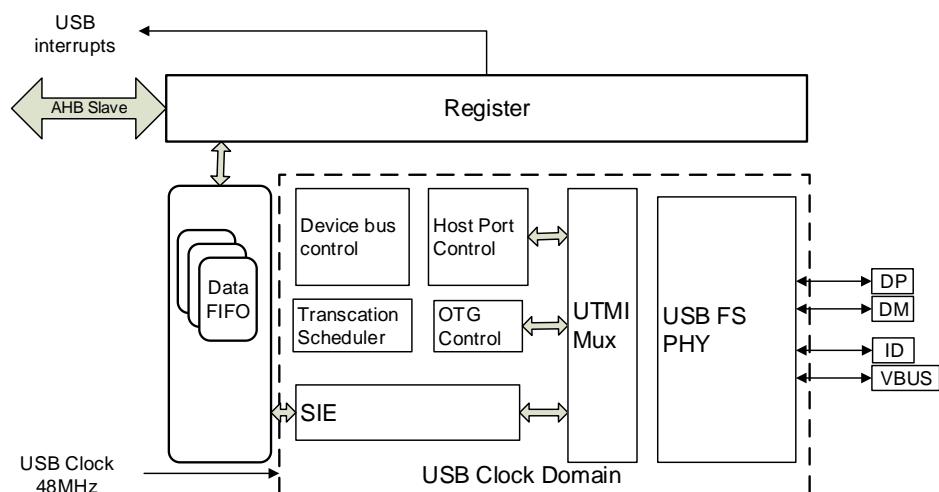
USB Full-Speed (USBFS) controller provides a USB-connection solution for portable devices. USBFS supports host and device modes, as well as OTG mode with HNP (Host Negotiation Protocol) and SRP (Session Request Protocol). USBFS contains a full-speed internal USB PHY and external PHY chip is not contained. USBFS supports all the four types of transfer (control, bulk, Interrupt and isochronous) which defined in USB 2.0 protocol.

24.2. Characteristics

- Supports USB 2.0 host mode at Full-Speed(12Mb/s) or Low-Speed(1.5Mb/s)
- Supports USB 2.0 device mode at Full-Speed(12Mb/s)
- Supports OTG protocol with HNP (Host Negotiation Protocol) and SRP (Session Request Protocol)
- Supports all the 4 types of transfer: control, bulk, interrupt and isochronous
- Includes a USB transaction scheduler in host mode to handle USB transaction request efficiently.
- Includes a 1.25KB FIFO RAM.
- Supports 8 channels in host mode.
- Includes 2 transmit FIFOs (periodic and non-periodic) and a receive FIFO (shared by all channels) in host mode.
- Includes 4 transmit FIFOs (one for each IN endpoint) and a receive FIFO (shared by all OUT endpoints) in device mode.
- Supports 4 OUT and 4 IN endpoints in device mode.
- Supports remote wakeup in device mode.
- Includes a Full-Speed USB PHY with OTG protocol supported.
- Time intervals of SOFs is dynamic adjustable in host mode.
- SOF pulse supports output to pad.
- Supports detecting ID pin level and VBUS voltage.
- Needs external component to supply power for connected USB device in host mode or OTG A-device mode.

24.3. Block diagram

Figure 24-1. USBFS block diagram



24.4. Signal description

Table 24-1. USBFS signal description

I/O port	Type	Description
VBUS	Input	Bus power port
DM	Input/Output	Differential D- port
DP	Input/Output	Differential D+ port
ID	Input	USB identification: Mini connector identification port

24.5. Function overview

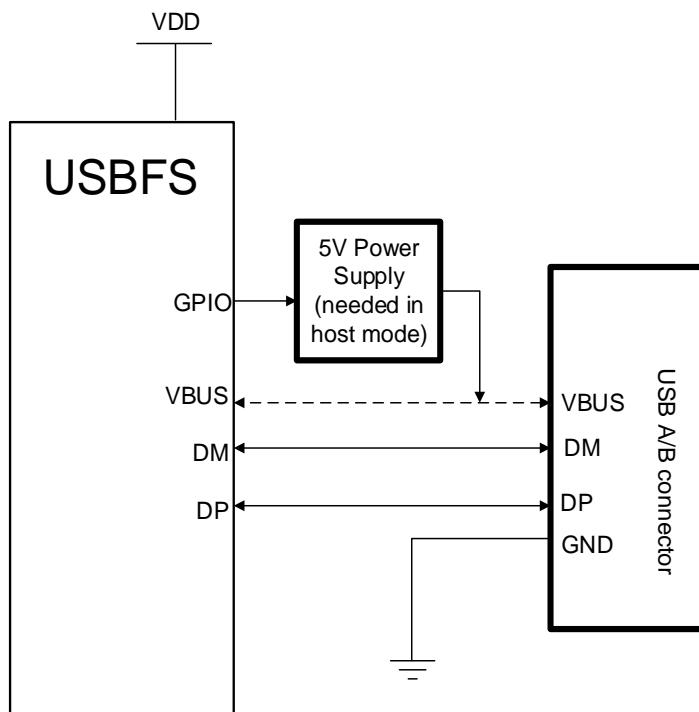
24.5.1. USBFS clocks and working modes

USBFS can operate as a host, a device or a DRD (Dual-role-Device), it contains an internal full-speed PHY. The maximum speed supported by USBFS is full-speed.

The internal PHY supports Full-Speed and Low-Speed in host mode, supports Full-speed in device mode, and supports OTG mode with HNP and SRP. The USB clock used by the USBFS should be 48MHz. The 48MHz USB clock is generated from internal clocks in system, and its source and divider factors are configurable in RCU.

The pull-up and pull-down resistors have already been integrated into the internal PHY and they could be controlled by USBFS automatically according to the current mode (host, device or OTG mode) and connection status. A typical connection is shown in [Figure 24-2 Connection with host or device mode](#)

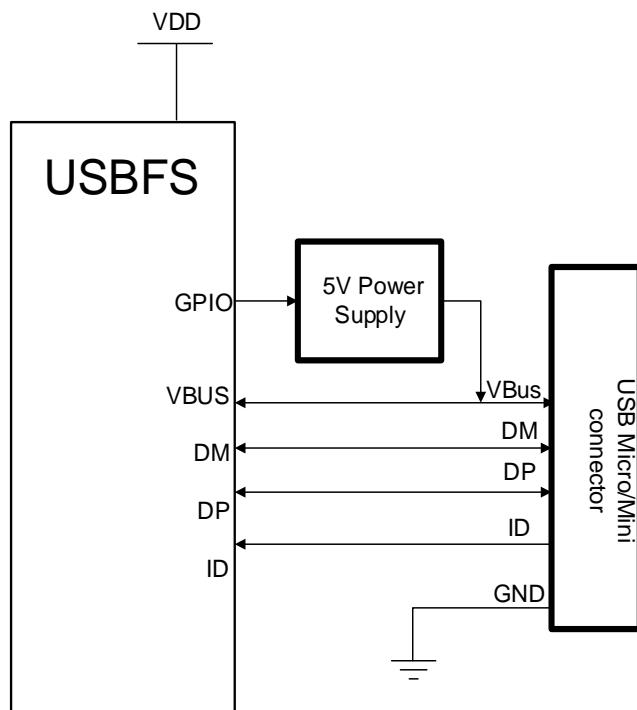
Figure 24-2. Connection with host or device mode



When USBFS works in host mode (FHM bit is set and FDM bit is cleared), the VBUS is 5V power detecting pin used for voltage detection defined in USB protocol. The internal PHY cannot supply 5V VBUS power and only has some voltage comparers, charge and dis-charge circuits on VBUS line. So if application needs VBUS power, an external power supply IC is needed. The VBUS connection between USBFS and the USB connector can be omitted in host mode, so USBFS doesn't detect the voltage level on VBUS pin and always assumes that the 5V power is present.

When USBFS works in device mode (FHM bit is cleared and FDM bit is set), the VBUS detection circuit is connected to a GPIO pin. USBFS continuously monitor the VBUS voltage by the GPIO pin and will immediately switch on the pull-up resistor on DP line once that the VBUS voltage rise above the needed valid value. This will cause a connection. If the VBUS voltage falls below the needed valid value, the pull-up resistor on DP line will be switched off and a disconnection will happen.

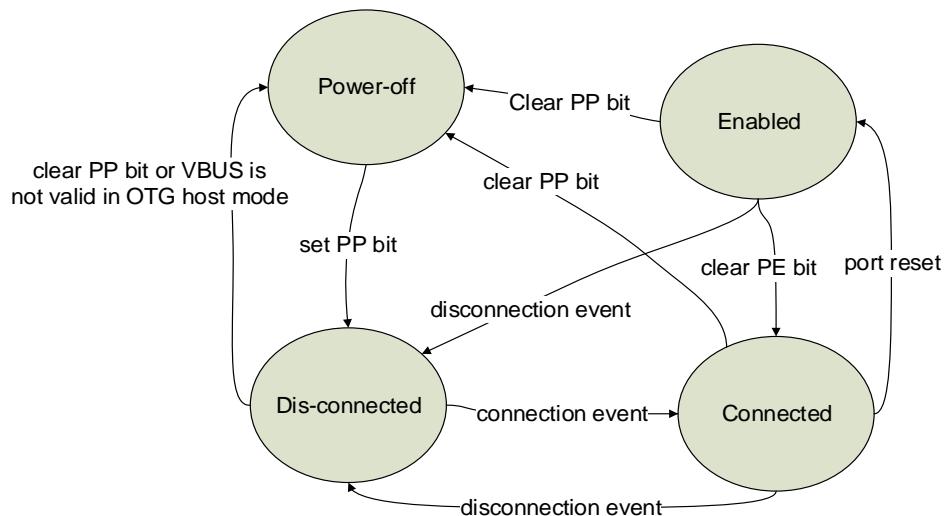
The OTG mode connection is described in the [Figure 24-3. Connection with OTG mode](#). When USBFS works in OTG mode, the FHM, FDM bits in USBFS_GUSBCS and VBUSIG bit in USBFS_GCCFG should be cleared. In this mode, the USBFS needs all the four pins: DM, DP, VBUS and ID, and needs to use several voltage comparers to monitor the voltage on these pins. USBFS also contains VBUS charge and discharge circuits to perform SRP request described in OTG protocol. The OTG A-device or B-device is decided by the level of ID pins. USBFS controls the pull-up or pull-down resistor during performing the HNP protocol.

Figure 24-3. Connection with OTG mode


24.5.2. USB host function

USB Host Port State

Host application may control state of the USB port via USBFS_HPCS register. After system initialization, the USB port stays at power-off state. After PP bit is set by software, the internal USB PHY is powered on, and the USB port changes into disconnected state. After a connection is detected, USB port changes into connected state. The USB port changes into enabled state after a port reset is performed on USB bus.

Figure 24-4. State transition diagram of host port


Connection, Reset and Speed identification

As a USB host, USBFS will trigger a connection flag for application after a connection is detected and will trigger a disconnection flag after a disconnection event.

PRST bit is used for USB reset sequence. Application may set this bit to start a USB reset and clear this bit to finish the USB reset. This bit only takes effect when port is at connected or enabled state.

The USBFS performs speed identification during connection, and the speed information will be reported in PS filed in USBFS_HPCS register. USBFS identifies the device speed by the voltage level of DM or DP. As described in USB protocol, full-speed device pulls up DP line while low-speed device pulls up DM line.

Suspend and resume

USBFS supports suspend state and resume operation. When USBFS port is at enabled state, writing 1 to PSP bit in USBFS_HPCS register will cause USBFS to enter suspend state. In suspend state, USBFS stops sending SOFs on USB bus and this will cause the connected USB device to enter suspend state after 3ms. Application can set the PREM bit in USBFS_HPCS register to start a resume sequence to wake up the suspended device and clear this bit to stop the resume sequence. The WKUPIF bit in USBFS_GINTF will be set and the USBFS wake up interrupt will be triggered if a host in suspend state detects a remote wakeup signal.

SOF generate

USBFS sends SOF tokens on USB bus in host mode. As described in USB 2.0 protocol, SOF packets are generated (by the host controller or hub transaction translator) every 1ms in full-speed links.

Each time after USBFS enters into enabled state, it will send the SOF packet periodically which the time is defined in USB 2.0 protocol. In addition, application may adjust the length of a frame by writing FRI filed in USBFS_HFT registers. The FRI bits define the number of USB clock cycles in a frame, so its value should be calculated based on the frequency of USB clock which is used by USBFS. The FRT filed bits show that the remaining clock cycles of the current frame and stop changing during suspend state.

USBFS is able to generate a pulse signal each SOF packet and output it to a pin. The pulse length is 12 HCLK cycle. If application desires to use this function, it needs to set SOFOEN bit in USBFS_GCCFG register and configure the related pin registers in GPIO.

USB Channels and Transactions

USBFS includes 8 independent channels in host mode. Each channel is able to communicate with an endpoint in USB device. The transfer type, direction, packet length and other information are all configured in channel related registers such as USBFS_HCHxCTL and USBFS_HCHxLEN.

USBFS supports all the four kinds of transfer types: control, bulk, interrupts and isochronous.

USB 2.0 protocol divides these transfers into 2 kinds: non-periodic transfer (control and bulk) and periodic transfer (interrupt and isochronous). Based on this, USBFS includes two request queues: periodic request queue and non-periodic request queue, to perform efficient transaction schedule. A request entry in a request queue described above may represent a USB transaction request or a channel operation request.

Application needs to write packet into data FIFO via AHB register interface if it wants to start an OUT transaction on USB bus. USBFS hardware will automatically generate a transaction request entry in request queue after the application writes a whole packet.

The request entries in request queue are processed in order by transaction control module. USBFS always tries to process periodic request queue firstly and secondly process non-periodic request queue.

After a start of frame, USBFS begins to process periodic queue until the queue is empty or bus time required by the current periodic request is not enough, and then process the non-periodic queue. This strategy ensures the bandwidth of periodic transactions in a frame. Each time the USBFS reads and pops a request entry from request queue. If this is a channel disable request, it immediately disables the channel and prepares to process the next entry.

If the current request is a transaction request and the USB bus time is enough for this transaction, USBFS will employ SIE to generate this transaction on USB bus.

When the required bus time for the current request is not enough in the current frame, and if this is a periodic request, USBFS stops processing the periodic queue and starts to process non-periodic request. If this is a non-periodic queue the USBFS will stop processing any queue and wait until the end of current frame.

24.5.3. USB device function

USB Device Connection

In device mode, USBFS stays at power-off state after initialization. After connecting to a USB host with 5V power supply through VBUS pin, USBFS enters into powered state. USBFS begins to switch on the pull-up resistor on DP line and thus, host side will detect a connection event.

Note: the VBUS pin must be connected to the PA9 for detecting the level.

Reset and Speed-Identification

The USB host always starts a USB reset when it detects a device connection, and USBFS in device mode will trigger a reset interrupt by hardware when it detects the reset event on USB bus.

After reset sequence, USBFS will trigger an ENUMF interrupt in USBFS_GINTF register and reports current enumerated device speed in ES bits in USBFS_DSTAT register, this bit field is always 11(full-speed).

As required by USB 2.0 protocol, USBFS doesn't support low-speed in device mode.

Suspend and Wake-up

A USB device will enter into suspend state when the USB bus stays at IDLE state and there is no change on data lines for 3ms. When USB device is in suspend state, most of its clock are closed to save power. The USB host is able to wake up the suspended device by generating a resume signal on USB bus. When USBFS detects the resume signal, the WKUPIF flag in USBFS_GINTF register will be set and the USBFS wake up interrupt will be triggered.

In suspend mode, USBFS is also able to remotely wake up the USB bus. Software may set RWKUP bit in USBFS_DCTL register to send a remote wake-up signal, and if remote wake-up is supported in USB host, the host will begin to send resume signal on USB bus.

Soft Disconnection

USBFS supports soft disconnection. After the device is powered on, USBFS will switch on the pull-up resistor on DP line so that the host can detect the connection. It is able to force a disconnection by setting the SD bit in USBFS_DCTL register. After the SD bit is set, USBFS will directly switch off the pull-up resistor, so that USB host will detect a disconnection on USB bus.

SOF tracking

When USBFS receives a SOF packet on USB bus, it will trigger a SOF interrupt and begin to count the bus time using local USB clock. The frame number of the current frame is reported in FNRSOF filed in USBFS_DSTAT register. When the USB bus time reaches EOF1 or EOF2 point (End of Frame, described in USB 2.0 protocol), USBFS will trigger an EOPFIF interrupt in USBFS_GINTF register. These flags and registers can be used to get current bus time and position information.

24.5.4. OTG function overview

USBFS supports OTG function described in OTG protocol 1.3, OTG function includes SRP and HNP protocols.

A-Device and B-Device

A-Device is an OTG capable USB device with a Standard-A or Micro-A plug inserted into its receptacle. The A-Device supplies power to VBUS and it is host at the start of a session. B-Device is an OTG capable USB device with a Standard-B, Micro-B or Mini-B plug inserted into its receptacle, or a captive cable ending being a Standard-A plug. The B-Device is a peripheral at the start of a session. USBFS uses the voltage level of ID pin to identify A-Device or B-Device. The ID status is reported in IDPS bit in USBFS_GOTGCS register. For the details of transfer states between A-Device and B-Device, please refer to OTG 1.3 protocol.

HNP

The Host Negotiation Protocol (HNP) allows the host function to be switched between two directly connected On-The-Go devices and eliminates the necessity of switching the cable connections for the change of control of communications between the devices. HNP will be initialized typically by the user or an application on the On-The-Go B-Device. HNP may only be implemented through the Micro-AB receptacle on a device.

Since On-The-Go devices have a Micro-AB receptacle, an On-The-Go device can default to being either a host or a device, depending that which type of plug (Micro-A plug for host, Micro-B plug for device) is inserted. By utilizing the Host Negotiation Protocol (HNP), an On-The-Go B-Device, which is the default device, may make a request to be a host. The process for the exchange of the role to a host is described in this section. This protocol eliminates the necessity of switching the cable connection for the change of the roles of the connected devices.

When USBFS is in OTG A-Device host mode and it wants to give up its host role, it may firstly set PSP bit in USBFS_HPCS register to make the USB bus enter in suspend status. Then, the B-Device will enter in suspend state 3ms later. If the B-Device wants to change to be a host, HNPREQ bit in USBFS_GOTGCS register should be set and the USBFS will begin to perform HNP protocol on bus, and at last, the result of HNP is reported in HNPS bit in USBFS_GOTGCS register. Besides, it is always available to get the current role (host or device) from COPM bit in USBFS_GINTF register.

SRP

The Session Request Protocol (SRP) allows a B-Device to request the A-Device to turn on VBUS and start a session. This protocol allows the A-Device, which may be battery powered, to conserve power by turning VBUS off when there is no bus activity while still providing a means for the B-Device to initiate bus activity. As described in OTG protocol, an OTG device must compare VBUS voltage with several threshold values and the compare result should be reported in ASV and BSV bits in USBFS_GOTGCS register.

Set SRPREQ bit in USBFS_GOTGCS register to start a SRP request when USBFS is in B-Device OTG mode and USBFS will generate a success flag SRPS in USBFS_GOTGCS register if the SRP request successfully.

When USBFS is in OTG A-Device mode and it has detected a SRP request from a B-Device, it sets a SESIF flag in USBFS_GINTF register. The 5V power supply for VBUS pin should be prepared to switch on after getting this flag.

24.5.5. Data FIFO

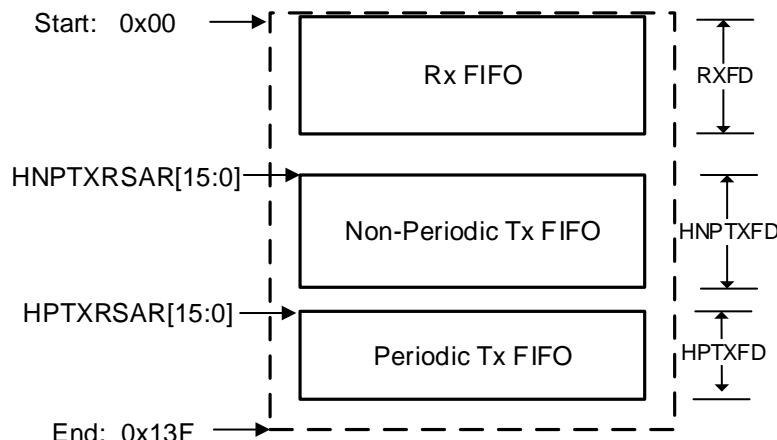
The USBFS contains a 1.25K bytes data FIFO for packet data storage. The data FIFO is implemented by using an internal SRAM in USBFS.

Host Mode

In host mode, the data FIFO space is divided into 3 parts: Rx FIFO for received packet, Non-Periodic Tx FIFO for non-period transmission packet and Periodic Tx FIFO for periodic

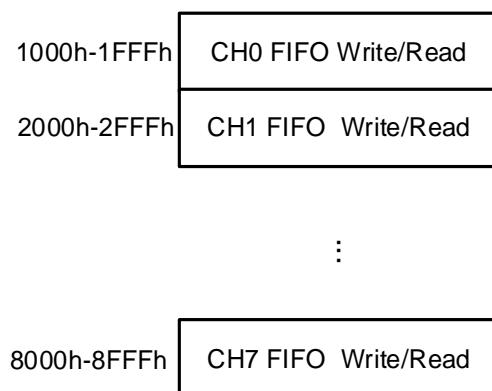
transmission packet. All IN channels shares the Rx FIFO for packets reception. All the periodic OUT channels share the periodic Tx FIFO to packets transmission. All the non-periodic OUT channels share the non-Periodic Tx FIFO for transmit packets. The size and start offset of these data FIFOs should be configured using these registers: USBFS_GRFLEN, USBFS_HNPTFLEN and USBFS_HPTFLEN. [Figure 24-5. HOST mode FIFO space in SRAM](#) describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

Figure 24-5. HOST mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. [Figure 24-6. Host mode FIFO access register map](#) describes the register memory area that the data FIFO can write. This area can be read by any channel data FIFO. The addresses in the figure are addressed in bytes. Each channel has its own FIFO access register space, although all Non-periodic channels share the same FIFO and all the Periodic channels also share the same FIFO. It is important for USBFS to know which channel the current pushed packet belongs to. Rx FIFO is also able to be accessed using USBFS_GRSTATR/USBFS_GRSTATP register.

Figure 24-6. Host mode FIFO access register map

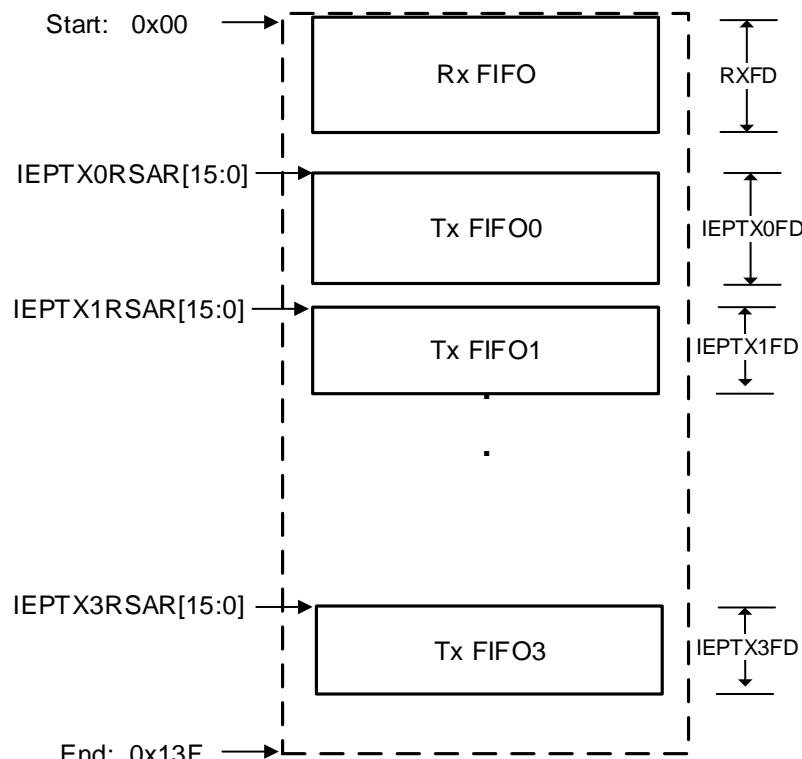


Device mode

In device mode, the data FIFO is divided into several parts: one Rx FIFO, and 4 Tx FIFOs

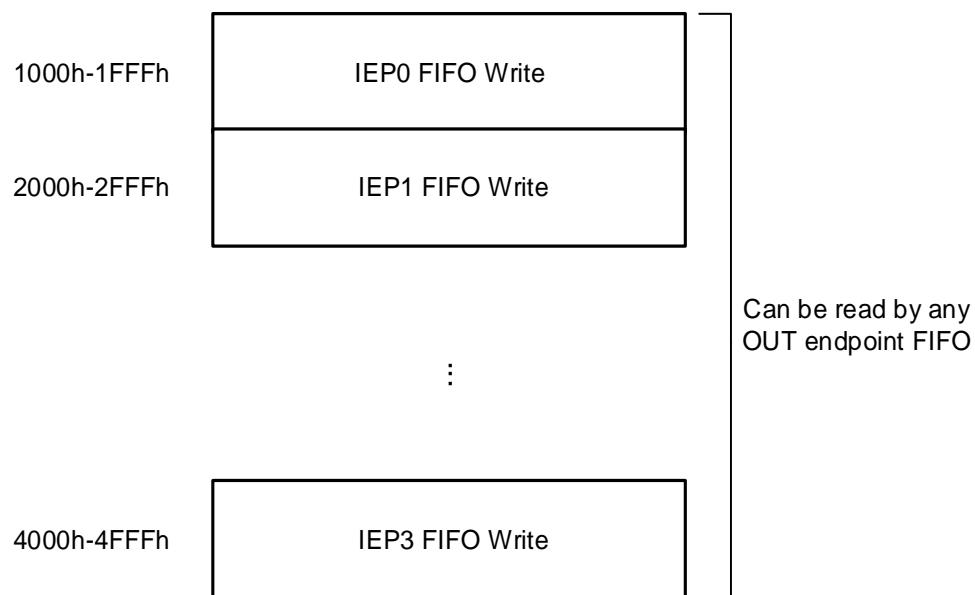
(one for each IN endpoint). All the OUT endpoints share the Rx FIFO for receiving packets. The size and start offset of these data FIFOs should be configured using USBFS_GRFLEN and USBFS_DIEPxTFLEN (x=0...3) registers. [Figure 24-7. Device mode FIFO space in SRAM](#) describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

Figure 24-7. Device mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. [Figure 24-8. Device mode FIFO access register map](#) describes the register memory area where the data FIFO can write. This area can be read by any endpoint FIFO. The addresses in the figure are addressed in bytes. Each endpoint has its own FIFO access register space. Rx FIFO is also able to be accessed using USBFS_GRSTATR/USBFS_GRSTATP register.

Figure 24-8. Device mode FIFO access register map



24.5.6. Operation guide

This section describes the advised operation guide for USBFS.

Host mode

Global register initialization sequence

1. Program USBFS_GAHBCS register according to application's demand, such as the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.
2. Program USBFS_GUSBCS register according to application's demand, such as the operation mode (host, device or OTG) and some parameters of OTG and USB protocols.
3. Program USBFS_GCCFG register according to application's demand.
4. Program USBFS_GRFLEN, USBFS_HNPTFLEN_DIEP0TFLEN and USBFS_HPTFLEN register to configure the data FIFOs according to application's demand.
5. Program USBFS_GINTEN register to enable Mode Fault and Host Port interrupt and set GINTEN bit in USBFS_GAHBCS register to enable global interrupt.
6. Program USBFS_HPCS register and set PP bit.
7. Wait for a device's connection, and once a device is connected, the connection interrupt PCD in USBFS_HPCS register will be triggered. Then set PRST bit to perform a port reset. Wait for at least 10ms and then clear PRST bit.
8. Wait PEDC interrupt in USBFS_HPCS register and then read PE bit to ensure that the port is successfully enabled. Read PS [1:0] bits to get the connected device's speed and then program USBFS_HFT register to change the SOF interval if needed.

Channel initialization and enable sequence

1. Program USBFS_HCHxCTL registers with desired transfer type, direction, packet size, etc. Ensure that CEN and CDIS bits keep cleared during configuration.
2. Program USBFS_HCHxINTEN register. Set the desired interrupt enable bits.
3. Program USBFS_HCHxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes number of all the transmitted or received packets in a transfer.

For OUT channel: If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBFS_HCHxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If software wants to send out a zero-length packet, it should program TLEN=0, PCNT=1.

For IN channel: Because the application doesn't know the actual received data size before the IN transaction finishes, TLEN can be set to a maximum possible value supported by Rx FIFO.

4. Set CEN bit in USBFS_HCHxCTL register to enable the channel.

Channel disable sequence

Software can disable the channel by setting both CEN and CDIS bits at the same time. USBFS will generate a channel disable request entry in request queue after the register setting operation. When the request entry reaches the top of request queue, it is processed by USBFS immediately:

For OUT channels, the specified channel will be disabled immediately. Then, a CH flag will be generated and the CEN and CDIS bits will be cleared by USBFS.

For IN channels, USBFS pushes a channel disable status entry into Rx FIFO. Software should then handle the Rx FIFO not empty event: read and pop this status entry, then, a CH flag will be generated and the CEN and CDIS bits will be cleared.

IN transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize the channel.
3. Enable the channel.
4. After the IN channel is enabled by software, USBFS generates an Rx request entry in the corresponding request queue.
5. When the Rx request entry reaches the top of the request queue, USBFS begins to process this request entry. If bus time for the IN transaction indicated by the request entry is enough, USBFS starts the IN transaction on USB bus.
6. If the IN transaction finishes successfully (ACK handshake received), USBFS pushes the

received data packet into the Rx FIFO and triggers ACK flag. Otherwise, the status flag (NAK) reports the transaction result.

7. If the IN transaction described in step 5 is successful and PCNT is larger than 1 in step2, return to step 3 and continues to receive the remaining packets. If the IN transaction described in step 5 is not successful, return to step 3 to re-receive the packet again.
8. After all the transactions in a transfer are successfully received on USB bus, USBFS pushes a TF status entry into the Rx FIFO on top of the last packet data. Thus after reading and popping all the received data packet, the TF status entry is need, USBFS generates TF flag to indicate that the transfer successfully finishes.
9. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

OUT transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize and enable the channel.
3. Write a packet into the channel's Tx FIFO (Periodic Tx FIFO or non-periodic Tx FIFO). After the whole packet data is written into the FIFO, USBFS generates a Tx request entry in the corresponding request queue and decreases the TLEN field in USBFS_HCHxLEN register by the written packet's size.
4. When the request entry reaches the top of the request queue, USBFS begins to process this request entry. If bus time for the transaction indicated by the request entry is enough, USBFS starts the OUT transaction on USB bus.
5. When the OUT transaction indicated by the request entry finishes on USB bus, PCNT in USBFS_HCHxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flag (NAK) reports the transaction result.
6. If the OUT transaction described in step 5 is successful and PCNT is larger than 1 in step2, return to step 3 and continues to send the remaining packets. If the OUT transaction described in step 5 is not successful, return to step 3 to resend the packet again.
7. After all the transactions in a transfer are successfully sent on USB bus, USBFS generates TF flag to indicate that the transfer successfully finishes.
8. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

Device mode

Global register initialization sequence

1. Program USBFS_GAHBCS register according to application's demand, such as the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.

2. Program USBFS_GUSBCS register according to application's demand, such as: the operation mode (host, device or OTG) and some parameters of OTG and USB protocols.
3. Program USBFS_GCCFG register according to application's demand.
4. Program USBFS_GRFLEN, USBFS_HNPTFLEN_DIEP0TFLEN, USBFS_DIEPxTFLEN register to configure the data FIFOs according to application's demand.
5. Program USBFS_GINTEN register to enable Mode Fault, Suspend, SOF, Enumeration Done and USB Reset interrupt and then, set GINTEN bit in USBFS_GAHBCS register to enable global interrupt.
6. Program USBFS_DCFG register according to application's demand, such as the device address, etc.
7. After the device is connected to a host, the host will perform port reset on USB bus and this will trigger the RST interrupt in USBFS_GINTF register.
8. Wait for ENUMF interrupt in USBFS_GINTF register.

Endpoint initialization and enable sequence

1. Program USBFS_DIEPxCTL or USBFS_DOEPxCTL register with desired transfer type, packet size, etc.
2. Program USBFS_DIEPINTEN or USBFS_DOEPINTEN register. Set the desired interrupt enable bits.
3. Program USBFS_DIEPxLEN or USBFS_DOEPxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes number of all the transmitted or received packets in a transfer.

For IN endpoint: If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBFS_DIEPxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If a zero-length packet is required to be sent, it should program TLEN=0, PCNT=1.

For OUT endpoint: Because the application doesn't know the actual received data size before the OUT transaction finishes, TLEN can be set to a maximum possible value supported by Rx FIFO.

4. Set EPEN bit in USBFS_DIEPxCTL or USBFS_DOEPxCTL register to enable the endpoint.

Endpoint disable sequence

The endpoint can be disabled anytime when the EPEN bit in USBFS_DIEPxCTL or USBFS_DOEPxCTL registers is cleared.

IN transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize and enable the IN endpoint.
3. Write packets into the endpoint's Tx FIFO. Each time a data packet is written into the FIFO, USBFS decreases the TLEN field in USBFS_DIEPxLEN register by the written packet's size.
4. When an IN token received, USBFS transmits the data packet, and after the transaction finishes on USB bus, PCNT in USBFS_DIEPxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flags reports the transaction result.
5. After all the data packets in a transfer are successfully sent on USB bus, USBFS generates TF flag to indicate that the transfer successfully finishes and disables the IN endpoint.

OUT transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize the endpoint and enable the endpoint.
3. When an OUT token received, USBFS receives the data packet or response with an NAK handshake based on the status of Rx FIFO and register configuration. If the transaction finishes successfully (USBFS receives and saves the data packet into Rx FIFO successfully and sends ACK handshake on USB bus), PCNT in USBFS_DOEPxLEN register is decreased by 1 and the ACK flag is triggered, otherwise, the status flags report the transaction result.
4. After all the data packets in a transfer are successfully received on USB bus, USBFS pushes a TF status entry into the Rx FIFO on top of the last packet data. Thus after reading and popping all the received data packet, the TF status entry is read, USBFS generates TF flag to indicate that the transfer successfully finishes and disables the OUT endpoint.

24.6. Interrupts

USBFS has two interrupts: global interrupt and wake-up interrupt.

The source flags of the global interrupt are readable in USBFS_GINTF register and are listed in [Table 24-2. USBFS global interrupt](#).

Table 24-2. USBFS global interrupt

Interrupt Flag	Description	Operation Mode
SEIF	Session interrupt	Host or device mode
DISCIF	Disconnect interrupt flag	Host Mode
IDPSC	ID pin status change	Host or device mode

Interrupt Flag	Description	Operation Mode
PTXFEIF	Periodic Tx FIFO empty interrupt flag	Host Mode
HCIF	Host channels interrupt flag	Host Mode
HPIF	Host port interrupt flag	Host Mode
ISOONCIF/PXNCIF	Periodic transfer Not Complete Interrupt flag /Isochronous OUT transfer Not Complete Interrupt Flag	Host or device mode
ISOINCIF	Isochronous IN transfer Not Complete Interrupt Flag	Device mode
OEPIF	OUT endpoint interrupt flag	Device mode
IEPIF	IN endpoint interrupt flag	Device mode
EOPFIF	End of periodic frame interrupt flag	Device mode
ISOOPDIF	Isochronous OUT packet dropped interrupt flag	Device mode
ENUMF	Enumeration finished	Device mode
RST	USB reset	Device mode
SP	USB suspend	Device mode
ESP	Early suspend	Device mode
GONAK	Global OUT NAK effective	Device mode
GNPINAK	Global IN Non-Periodic NAK effective	Device mode
NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag	Host Mode
RXFNEIF	Rx FIFO non-empty interrupt flag	Host or device mode
SOF	Start of frame	Host or device mode
OTGIF	OTG interrupt flag	Host or device mode
MFIF	Mode fault interrupt flag	Host or device mode

Wake-up interrupt can be triggered when USBFS is in suspend state, even when the USBFS's clocks are stopped. The source of the wake-up interrupt is WKUPIF bit in USBHS_GINTF register.

24.7. Register definition

USBFS secure base address: 0x5900 0000
 USBFS non-secure base address: 0x4900 0000

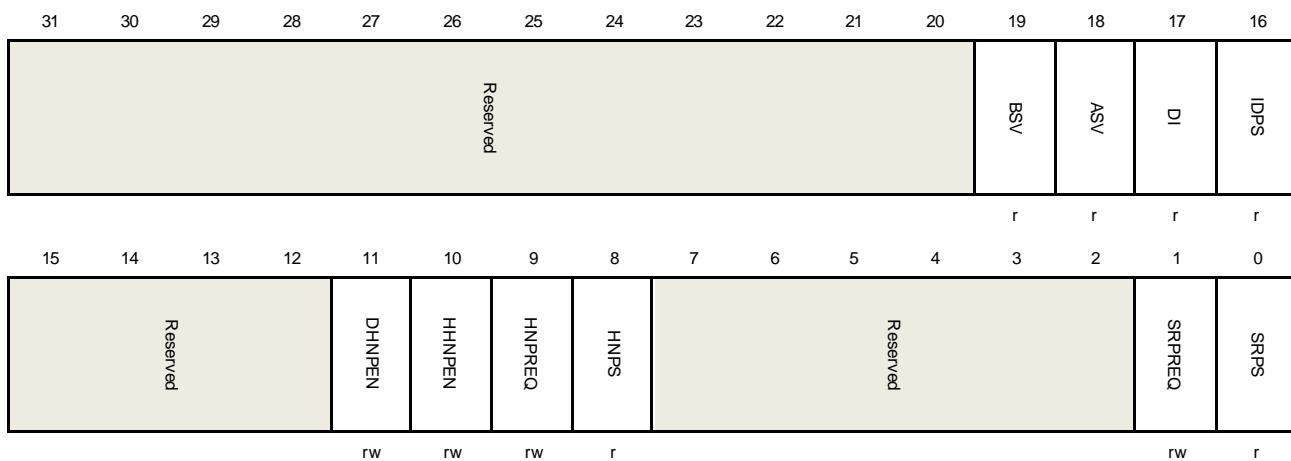
24.7.1. Global control and status registers

Global OTG control and status register (USBFS_GOTGCS)

Address offset: 0x0000

Reset value: 0x0000 0800

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	BSV	B-Session Valid (described in OTG protocol). 0: Vbus voltage level of a OTG B-Device is below VBSESSVLD 1: Vbus voltage level of a OTG B-Device is above VBSESSVLD Note: Only accessible in OTG B-Device mode.
18	ASV	A- Session valid A-host mode transceiver status. 0: Vbus voltage level of a OTG A-Device is below VASESSVLD 1: Vbus voltage level of a OTG A-Device is above VASESSVLD The A-Device is the default host at the start of a session. Note: Only accessible in OTG A-Device mode.
17	DI	Debounce interval Debounce interval of a detected connection. 0: Indicates the long debounce interval , when a plug-on and connection occurs on USB bus

		1: Indicates the short debounce interval, when a soft connection is used in HNP protocol. Note: Only accessible in host mode.
16	IDPS	<p>ID pin status</p> <p>Voltage level of connector ID pin</p> <p>0: USBFS is in A-Device mode</p> <p>1: USBFS is in B-Device mode</p> <p>Note: Accessible in both device and host modes.</p>
15:12	Reserved	Must be kept at reset value
11	DHNPEN	<p>Device HNP enable</p> <p>Enable the HNP function of a B-Device. If this bit is cleared, USBFS doesn't start HNP protocol when application set HNPREQ bit in USBFS_GOTGCS register.</p> <p>0: HNP function is not enabled.</p> <p>1: HNP function is enabled</p> <p>Note: Only accessible in device mode.</p>
10	HHPEN	<p>Host HNP enable</p> <p>Enable the HNP function of an A-Device. If this bit is cleared, USBFS doesn't response to the HNP request from B-Device.</p> <p>0: HNP function is not enabled.</p> <p>1: HNP function is enabled</p> <p>Note: Only accessible in host mode.</p>
9	HNPREQ	<p>HNP request</p> <p>This bit is set by software to start a HNP on the USB. This bit can be cleared when HNPEND bit in USBFS_GOTGINTF register is set, by writing zero to it, or clearing the HNPEND bit in USBFS_GOTGINTF register.</p> <p>0: Don't send HNP request</p> <p>1: Send HNP request</p> <p>Note: Only accessible in device mode.</p>
8	HNPS	<p>HNP successes</p> <p>This bit is set by the core when HNP succeeds, and this bit is cleared when HNPREQ bit is set.</p> <p>0: HNP fails</p> <p>1: HNP succeeds</p> <p>Note: Only accessible in device mode.</p>
7:2	Reserved	Must be kept at reset value
1	SRPREQ	<p>SRP request</p> <p>This bit is set by software to start a SRP on the USB. This bit can be cleared when SRPEND bit in USBFS_GOTGINTF register is set, by writing zero to it, or clearing the SRPEND bit in USBFS_GOTGINTF register.</p> <p>0: No session request</p>

1: Session request

Note: Only accessible in device mode.

0	SRPS	SRP success
		This bit is set by the core when SRP succeeds, and this bit is cleared when SRPREQ bit is set.
	0: SRP fails	
	1: SRP succeeds	

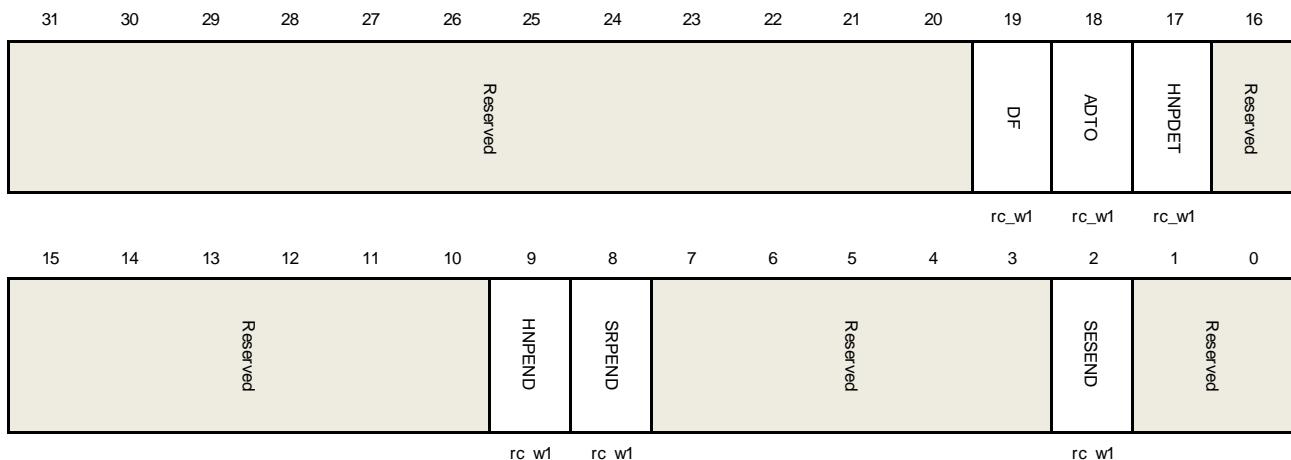
Note: Only accessible in device mode.

Global OTG interrupt flag register (USBFS_GOTGINTF)

Address offset: 0x0004

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	DF	<p>Debounce finish</p> <p>Set by USBFS when the debounce during device connection is done.</p> <p>Note: Only accessible in host mode.</p>
18	ADTO	<p>A-Device timeout</p> <p>Set by USBFS when the A-Device's waiting for a B-Device' connection has timed out.</p> <p>Note: Accessible in both device and host modes.</p>
17	HNPDET	<p>Host negotiation request detected</p> <p>Set by USBFS when A-Device detects a HNP request.</p> <p>Note: Accessible in both device and host modes.</p>
16:10	Reserved	Must be kept at reset value

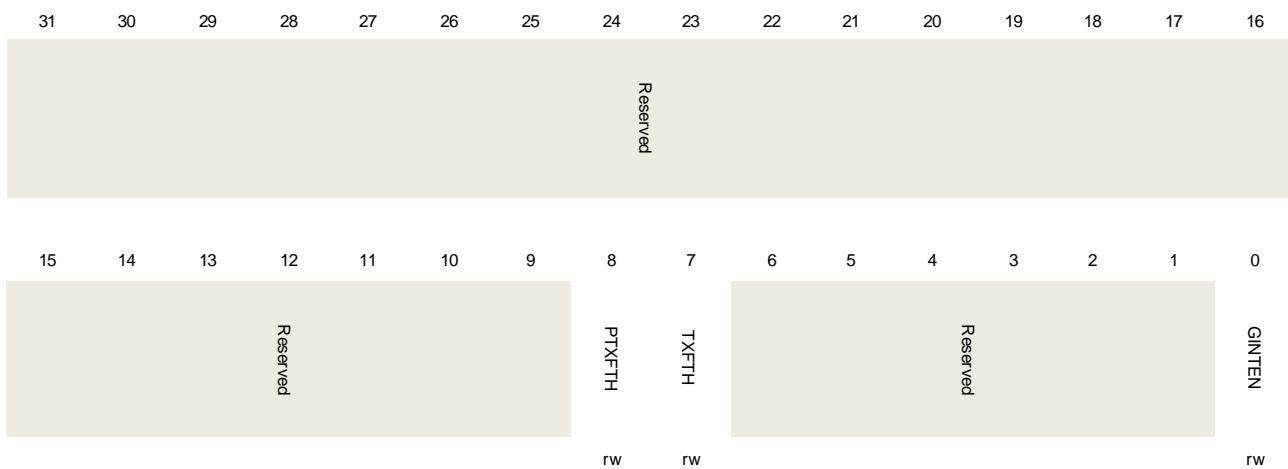
9	HNPEND	HNP end Set by the core when a HNP ends. Read the HNPS in USBFS_GOTGCS register to get the result of HNP. Note: Accessible in both device and host modes.
8	SRPEND	SRPEND Set by the core when a SRP ends. Read the SRPS in USBFS_GOTGCS register to get the result of SRP. Note: Accessible in both device and host modes.
7:3	Reserved	Must be kept at reset value
2	SESEND	Session end Set by the core when VBUS voltage is below Vb_ses_vld.
1:0	Reserved	Must be kept at reset value

Global AHB control and status register (USBFS_GAHBCS)

Address offset: 0x0008

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	PTXFTH	Periodic Tx FIFO threshold 0: PTXFEIF will be triggered when the periodic transmit FIFO is half empty 1: PTXFEIF will be triggered when the periodic transmit FIFO is completely empty Note: Only accessible in host mode.
7	TXFTH	Tx FIFO threshold Device mode: 0: TXFEIF will be triggered when the IN endpoint transmit FIFO is half empty

1: TXFEIF will be triggered when the IN endpoint transmit FIFO is completely empty

Host mode:

0: NPTXFEIF will be triggered when the non-periodic transmit FIFO is half empty

1: NPTXFEIF will be triggered when the non-periodic transmit FIFO is completely empty

6: 1 Reserved Must be kept at reset value

0 GINTEN Global interrupt enable

0: Global interrupt is not enabled.

1: Global interrupt is enabled.

Note: Accessible in both device and host modes.

Global USB control and status register (USBFS_GUSBCS)

Address offset: 0x000C

Reset value: 0x0000 0A80

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	FDM	FHM								Reserved					
	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			UTT[3:0]		HNPGEN	SRPCEN			Reserved			TOC[2:0]			
			rw		r/rw	r/rw						rw			

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	FDM	Force device mode Setting this bit will force the core to device mode irrespective of the USBFS ID input pin. 0: Normal mode 1: Device mode The application must wait at least 25 ms for the change taking effect after setting the force bit. Note: Accessible in both device and host modes.
29	FHM	Force host mode Setting this bit will force the core to host mode irrespective of the USBFS ID input

		pin.
		0: Normal mode
		1: Host mode
		The application must wait at least 25 ms for the change taking effect after setting the force bit.
		Note: Accessible in both device and host modes.
28:14	Reserved	Must be kept at reset value
13:10	UTT[3:0]	USB turnaround time Turnaround time in PHY clocks. Note: Only accessible in device mode.
9	HNPSEN	HNP capability enable Controls whether the HNP capability is enabled 0: HNP capability is disabled 1: HNP capability is enabled Note: Accessible in both device and host modes.
8	SRPCEN	SRP capability enable Controls whether the SRP capability is enabled 0: SRP capability is disabled 1: SRP capability is enabled Note: Accessible in both device and host modes.
7:3	Reserved	Must be kept at reset value
2:0	TOC[2:0]	Timeout calibration USBFS always uses time-out value required in USB 2.0 when waiting for a packet. Application may use TOC [2:0] to add the value in terms of PHY clock. (The frequency of PHY clock is 48MHz.).

Global reset control register (USBFS_GRSTCTL)

Address offset: 0x0010

Reset value: 0x8000 0000

The application uses this register to reset various hardware features inside the core.

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

Reserved

Reserved	TXFNUM[4:0]	rw	TXFF	RXFF	Reserved	HFCRST	HCSRST	CSRST
			rs	rs		rs	rs	rs

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
10:6	TXFNUM[4:0]	<p>Tx FIFO number</p> <p>Indicates which Tx FIFO will be flushed when TXFF bit in the same register is set.</p> <p>Host Mode:</p> <ul style="list-style-type: none"> 00000: Only non-periodic Tx FIFO is flushed 00001: Only periodic Tx FIFO is flushed 1XXXX: Both periodic and non-periodic Tx FIFOs are flushed Other: Non data FIFO is flushed <p>Device Mode:</p> <ul style="list-style-type: none"> 00000: Only Tx FIFO0 is flushed 00001: Only Tx FIFO1 is flushed ... 00011: Only Tx FIFO3 is flushed 1XXXX: All Tx FIFOs are flushed Other: Non data FIFO is flushed
5	TXFF	<p>Tx FIFO flush</p> <p>Application set this bit to flush data Tx FIFOs and TXFNUM[4:0] bits decide the FIFO number to be flushed. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS.</p> <p>Note: Accessible in both device and host modes.</p>
4	RXFF	<p>Rx FIFO flush</p> <p>Application set this bit to flush data Rx FIFO. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS.</p> <p>Note: Accessible in both device and host modes.</p>
3	Reserved	Must be kept at reset value
2	HFCRST	<p>Host frame counter reset</p> <p>Set by the application to reset the frame number counter in USBFS. After this bit is set, the frame number of the following SOF returns to 0. Hardware automatically clears this bit after the reset process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS.</p> <p>Note: Only accessible in host mode.</p>
1	HCSRST	HCLK soft reset

Set by the application to reset AHB clock domain circuit.

Hardware automatically clears this bit after the reset process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS.

Note: Accessible in both device and host modes.

0	CSRST	Core soft reset
Resets the AHB and USB clock domains circuits, as well as most of the registers.		

Global interrupt flag register (USBFS_GINTF)

Address offset: 0x0014

Reset value: 0x0400 0021

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
WKUPIF	SESIF	DISCIF	IDPSC	Reserved.	PTXFEIF	HCIIF	HPIIF	Reserved	PXNCIF/SOONCIF	ISOINCIF	OEPIF	IEPIF	Reserved	Reserved	Reserved
rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	r	rc_w1	rc_w1	r	r	r	r	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPIF	ISOOPDIF	ENUMF	RST	SP	ESP	Reserved	GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MIFIF	COPM	
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1	r	rc_w1	r	r

Bits	Fields	Descriptions
31	WKUPIF	Wakeup interrupt flag This interrupt is triggered when a resume signal (in device mode) or a remote wakeup signal (in host mode) is detected on the USB. Note: Accessible in both device and host modes.
30	SESIF	Session interrupt flag This interrupt is triggered when a SRP is detected (in A-Device mode) or V _{BUS} becomes valid for a B- Device (in B-Device mode). Note: Accessible in both device and host modes.
29	DISCIF	Disconnect interrupt flag This interrupt is triggered after a device disconnection. Note: Only accessible in host mode.
28	IDPSC	ID pin status change Set by the core when ID status changes.

		Note: Accessible in both device and host modes.
27	Reserved	Must be kept at reset value
26	PTXFEIF	<p>Periodic Tx FIFO empty interrupt flag</p> <p>This interrupt is triggered when the periodic transmit FIFO is either half or completely empty. The threshold is determined by the periodic Tx FIFO empty level bit (PTXFTH) in the USBFS_GAHBCS register.</p> <p>Note: Only accessible in host mode.</p>
25	HCIF	<p>Host channels interrupt flag</p> <p>Set by USBFS when one of the channels in host mode has raised an interrupt. First read USBFS_HACHINT register to get the channel number, and then read the corresponding USBFS_HCHxINTF register to get the flags of the channel that cause the interrupt. This bit will be automatically cleared after the respective channel's flags which cause channel interrupt are cleared.</p> <p>Note: Only accessible in host mode.</p>
24	HPIF	<p>Host port interrupt flag</p> <p>Set by the core when USBFS detects that port status changes in host mode. Software should read USBFS_HPCS register to get the source of this interrupt. This bit will be automatically cleared after the flags that causing a port interrupt are cleared.</p> <p>Note: Only accessible in host mode.</p>
23:22	Reserved	Must be kept at reset value
21	PXNCIF	<p>Periodic transfer Not Complete Interrupt flag</p> <p>USBFS sets this bit when there are periodic transactions for current frame not completed at the end of frame. (Host mode)</p>
	ISOONCIF	<p>Isochronous OUT transfer Not Complete Interrupt Flag</p> <p>At the end of a periodic frame (defined by EOPFT bit in USBFS_DCFG), USBFS will set this bit if there are still isochronous OUT endpoints for that not completed transactions. (Device Mode)</p>
20	ISOINCIF	<p>Isochronous IN transfer Not Complete Interrupt Flag</p> <p>At the end of a periodic frame (defined by EOPFT [1:0] bits in USBFS_DCFG), USBFS will set this bit if there are still isochronous IN endpoints for that not completed transactions. (Device Mode)</p> <p>Note: Only accessible in device mode.</p>
19	OEPIF	<p>OUT endpoint interrupt flag</p> <p>Set by USBFS when one of the OUT endpoints in device mode has raised an interrupt. Software should first read USBFS_DAEPINT register to get the device number, and then read the corresponding USBFS_DOEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared.</p>

		Note: Only accessible in device mode.
18	IEPIF	<p>IN endpoint interrupt flag</p> <p>Set by USBFS when one of the IN endpoints in device mode has raised an interrupt. Software should first read USBFS_DA EPINT register to get the device number, and then read the corresponding USBFS_DIEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared.</p> <p>Note: Only accessible in device mode.</p>
17:16	Reserved	Must be kept at reset value
15	EOPFIF	<p>End of periodic frame interrupt flag</p> <p>When USB bus time in a frame reaches the value defined by EOPFT [1:0] bits in USBFS_DCFG register, USBFS sets this flag.</p> <p>Note: Only accessible in device mode.</p>
14	ISOOPDIF	<p>Isochronous OUT packet dropped interrupt flag</p> <p>USBFS set this bit if it receives an isochronous OUT packet but cannot save it into Rx FIFO because the FIFO doesn't have enough space.</p> <p>Note: Only accessible in device mode.</p>
13	ENUMF	<p>Enumeration finished</p> <p>USBFS sets this bit after the speed enumeration finishes. Read USBFS_DSTAT register to get the current device speed.</p> <p>Note: Only accessible in device mode.</p>
12	RST	<p>USB reset</p> <p>USBFS sets this bit when it detects a USB reset signal on bus.</p> <p>Note: Only accessible in device mode.</p>
11	SP	<p>USB suspend</p> <p>USBFS sets this bit when it detects that the USB bus is idle for 3 ms and enters suspend state.</p> <p>Note: Only accessible in device mode.</p>
10	ESP	<p>Early suspend</p> <p>USBFS sets this bit when it detects that the USB bus is idle for 3 ms.</p> <p>Note: Only accessible in device mode.</p>
9:8	Reserved	Must be kept at reset value
7	GONAK	<p>Global OUT NAK effective</p> <p>Write 1 to SGONAK bit in the USBFS_DCTL register and USBFS will set GONA K flag after the writing to SGONAK takes effect.</p> <p>Note: Only accessible in device mode.</p>
6	GNPINAK	<p>Global Non-Periodic IN NAK effective</p> <p>Write 1 to SGINAK bit in the USBFS_DCTL register and USBFS will set GNPINAK</p>

		flag after the writing to SGINAK takes effect. Note: Only accessible in device mode.
5	NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag This interrupt is triggered when the non-periodic transmit FIFO is either half or completely empty. The threshold is determined by the non-periodic Tx FIFO empty level bit (TXFTH) in the USBFS_GAHBCS register. Note: Only accessible in host mode.
4	RXFNEIF	Rx FIFO non-empty interrupt flag USBFS sets this bit when there is at least one packet or status entry in the Rx FIFO. Note: Accessible in both host and device modes.
3	SOF	Start of frame Host Mode: USBFS sets this bit when it prepares to transmit a SOF or Keep-Alive on USB bus. Software can clear this bit by writing 1. Device Mode: USBFS sets this bit after it receives a SOF token. The application can read the Device Status register to get the current frame number. Software can clear this bit by writing 1. Note: Accessible in both host and device modes.
2	OTGIF	OTG interrupt flag USBFS sets this bit when the flags in USBFS_GOTGINTE register generate an interrupt. Software should read USBFS_GOTGINTE register to get the source of this interrupt. This bit is cleared after the flags in USBFS_GOTGINTE causing this interrupt are cleared. Note: Accessible in both host and device modes.
1	MFIF	Mode fault interrupt flag USBFS sets this bit when software operates host-only register in device mode, or operates device-mode in host mode. These fault operations won't take effect. Note: Accessible in both host and device modes.
0	COPM	Current operation mode 0: Device mode 1: Host mode Note: Accessible in both host and device modes.

Global interrupt enable register (USBFS_GINTEN)

Address offset: 0x0018

Reset value: 0x0000 0000

This register works with the global interrupt flag register (USBFS_GINTF) to interrupt the application. When an interrupt enable bit is disabled, the interrupt associated with that bit is not generated. However, the global Interrupt flag register bit corresponding to that interrupt is still set.

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
WKUPIE	SEIE	DISCIE	IDPSCIE	Reserved.	PTXFEIE	HCIE	HPIE	Reserved	PXNCIE/ ISOONCIE	ISOINCIE	OEPIE	IEPIE	Reserved	Reserved	
rw	rw	rw	rw		rw	rw	r		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIE	ISOOPDIE	ENUMFIE	RSTIE	SPIE	ESPIE	Reserved	GONAKIE	GNPINAKIE	NPTXFEIE	RXFNEIE	SOFIE	OTGIE	MFIE	Reserved	
rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31	WKUPIE	Wakeup interrupt enable 0: Disable wakeup interrupt 1: Enable wakeup interrupt Note: Accessible in both host and device modes.
30	SEIE	Session interrupt enable 0: Disable session interrupt 1: Enable session interrupt Note: Accessible in both host and device modes.
29	DISCIE	Disconnect interrupt enable 0: Disable disconnect interrupt 1: Enable disconnect interrupt Note: Only accessible in device mode.
28	IDPSCIE	ID pin status change interrupt enable 0: Disable connector ID pin status interrupt 1: Enable connector ID pin status interrupt Note: Accessible in both host and device modes.
27	Reserved	Must be kept at reset value
26	PTXFEIE	Periodic Tx FIFO empty interrupt enable 0: Disable periodic Tx FIFO empty interrupt 1: Enable periodic Tx FIFO empty interrupt Note: Only accessible in host mode.
25	HCIE	Host channels interrupt enable 0: Disable host channels interrupt 1: Enable host channels interrupt Note: Only accessible in host mode.

24	HPIE	Host port interrupt enable 0: Disable host port interrupt 1: Enable host port interrupt Note: Only accessible in host mode.
23:22	Reserved	Must be kept at reset value
21	PXNCIE	Periodic transfer not complete Interrupt enable 0: Disable periodic transfer not complete interrupt 1: Enable periodic transfer not complete interrupt Note: Only accessible in host mode.
	ISOONCIE	Isochronous OUT transfer not complete interrupt enable 0: Disable isochronous OUT transfer not complete interrupt 1: Enable isochronous OUT transfer not complete interrupt Note: Only accessible in device mode.
20	ISOINCIE	Isochronous IN transfer not complete interrupt enable 0: Disable isochronous IN transfer not complete interrupt 1: Enable isochronous IN transfer not complete interrupt Note: Only accessible in device mode.
19	OEPIE	OUT endpoints interrupt enable 0: Disable OUT endpoints interrupt 1: Enable OUT endpoints interrupt Note: Only accessible in device mode.
18	IEPIE	IN endpoints interrupt enable 0: Disable IN endpoints interrupt 1: Enable IN endpoints interrupt Note: Only accessible in device mode.
17:16	Reserved	Must be kept at reset value
15	EOPFIE	End of periodic frame interrupt enable 0: Disable end of periodic frame interrupt 1: Enable end of periodic frame interrupt Note: Only accessible in device mode.
14	ISOOPDIE	Isochronous OUT packet dropped interrupt enable 0: Disable isochronous OUT packet dropped interrupt 1: Enable isochronous OUT packet dropped interrupt Note: Only accessible in device mode.
13	ENUMFIE	Enumeration finish enable 0: Disable enumeration finish interrupt 1: Enable enumeration finish interrupt Note: Only accessible in device mode.
12	RSTIE	USB reset interrupt enable

		0: Disable USB reset interrupt 1: Enable USB reset interrupt Note: Only accessible in device mode.
11	SPIE	USB suspend interrupt enable 0: Disable USB suspend interrupt 1: Enable USB suspend interrupt Note: Only accessible in device mode.
10	ESPIE	Early suspend interrupt enable 0: Disable early suspend interrupt 1: Enable early suspend interrupt Note: Only accessible in device mode.
9:8	Reserved	Must be kept at reset value
7	GONAKIE	Global OUT NAK effective interrupt enable 0: Disable global OUT NAK interrupt 1: Enable global OUT NAK interrupt Note: Only accessible in device mode.
6	GNPINAKIE	Global non-periodic IN NAK effective interrupt enable 0: Disable global non-periodic IN NAK effective interrupt 1: Enable global non-periodic IN NAK effective interrupt Note: Only accessible in device mode.
5	NPTXFEIE	Non-periodic Tx FIFO empty interrupt enable 0: Disable non-periodic Tx FIFO empty interrupt 1: Enable non-periodic Tx FIFO empty interrupt Note: Only accessible in Host mode.
4	RXFNEIE	Receive FIFO non-empty interrupt enable 0: Disable receive FIFO non-empty interrupt 1: Enable receive FIFO non-empty interrupt Note: Accessible in both device and host modes.
3	SOFIE	Start of frame interrupt enable 0: Disable start of frame interrupt 1: Enable start of frame interrupt Note: Accessible in both device and host modes.
2	OTGIE	OTG interrupt enable 0: Disable OTG interrupt 1: Enable OTG interrupt Note: Accessible in both device and host modes.
1	MFIE	Mode fault interrupt enable 0: Disable mode fault interrupt 1: Enable mode fault interrupt

Note: Accessible in both device and host modes.

0	Reserved	Must be kept at reset value
---	----------	-----------------------------

Global receive status read/receive status read and pop registers (USBFS_GRSTATR/USBFS_GRSTATP)

Address offset for Read: 0x001C

Address offset for Pop: 0x0020

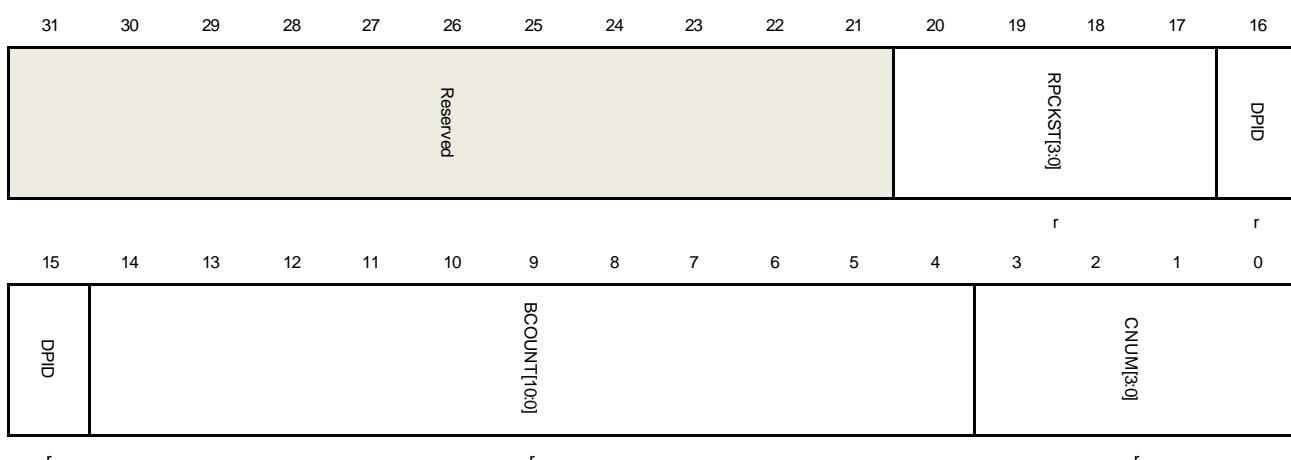
Reset value: 0x0000 0000

A read to the receive status read register returns the entry of the top of the Rx FIFO. A read to the Receive status read and pop register additionally pops the top entry out of the Rx FIFO.

The entries in RxFIFO have different meanings in host and device modes. Software should only read this register after when Receive FIFO non-empty interrupt flag bit of the global interrupt flag register (RXFNEIF bit in USBFS_GINTF) is triggered.

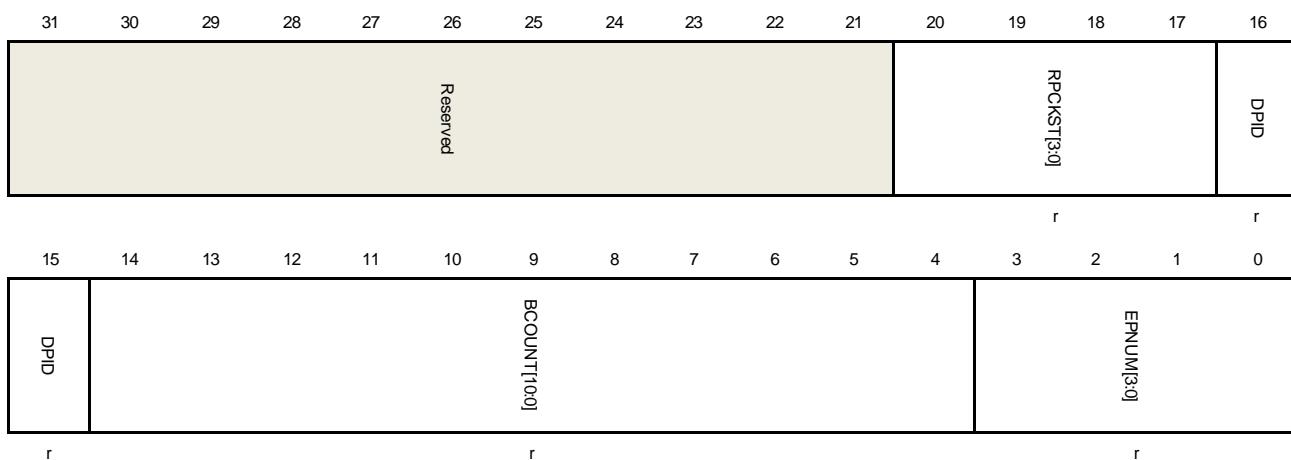
This register has to be accessed by word (32-bit)

Host mode:



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value
20:17	RPCKST[3:0]	Received packet status 0010: IN data packet received 0011: IN transfer completed (generates an interrupt if popped) 0101: Data toggle error (generates an interrupt if popped) 0111: Channel halted (generates an interrupt if popped) Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received packet 00: DATA0 10: DATA1

		01: DATA2
		11: MDATA
14:4	BCOUNT[10:0]	Byte count The byte count of the received IN data packet.
3:0	CNUM[3:0]	Channel number The channel number to which the current received packet belongs.

Device mode:


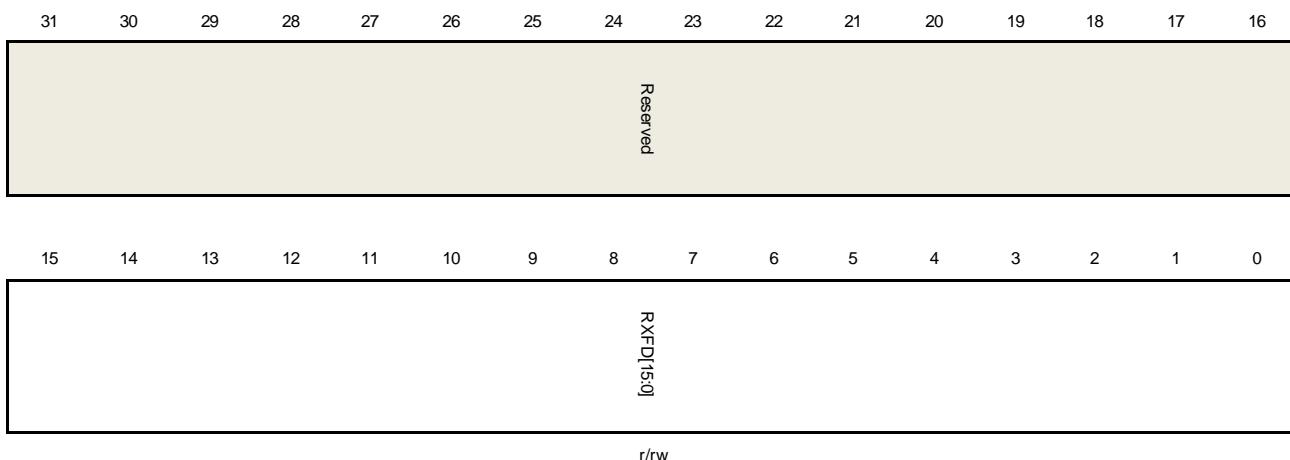
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value
20:17	RPCKST[3:0]	Received packet status 0001: Global OUT NAK (generates an interrupt) 0010: OUT data packet received 0011: OUT transfer completed (generates an interrupt) 0100: SETUP transaction completed (generates an interrupt) 0110: SETUP data packet received Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received OUT data packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA
14:4	BCOUNT[10:0]	Byte count The byte count of the received data packet.
3:0	EPNUM[3:0]	Endpoint number The endpoint number to which the current received packet belongs.

Global receive FIFO length register (USBFS_GRFLEN)

Address offset: 0x024

Reset value: 0x0000 0200

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



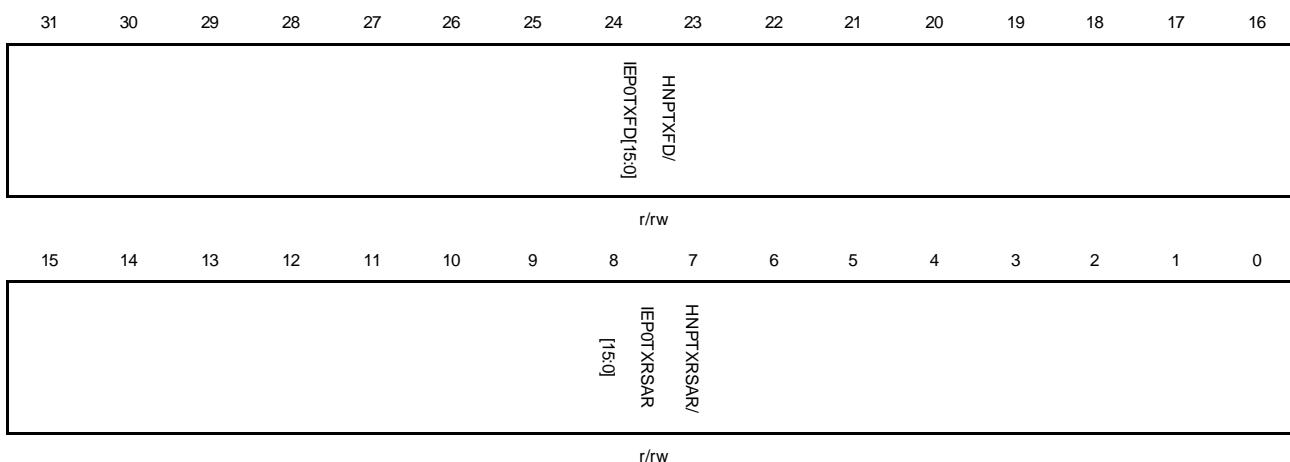
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	RXFD[15:0]	Rx FIFO depth In terms of 32-bit words. $1 \leq RXFD \leq 1024$

Host non-periodic transmit FIFO length register /Device IN endpoint 0 transmit
FIFO length (USBFS_HNPTFLEN _DIEP0TFLEN)

Address offset: 0x028

Reset value: 0x0200 0200

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



Host Mode:

Bits	Fields	Descriptions
31:16	HNPTXFD[15:0]	Host Non-periodic Tx FIFO depth In terms of 32-bit words. $1 \leq HNPTXFD \leq 1024$
15:0	HNPTXRSAR[15:0]	Host Non-periodic Tx RAM start address The start address for non-periodic transmit FIFO RAM is in term of 32-bit words.

Device Mode:

Bits	Fields	Descriptions
31:16	IEP0TXF[15:0]	IN Endpoint 0 Tx FIFO depth In terms of 32-bit words. $16 \leq IEP0TXF \leq 140$
15:0	IEP0TXRSA R[15:0]	IN Endpoint 0 TX RAM start address The start address for endpoint0 transmit FIFO RAM is in term of 32-bit words.

Host non-periodic transmit FIFO/queue status register (USBFS_HNPTFQSTAT)

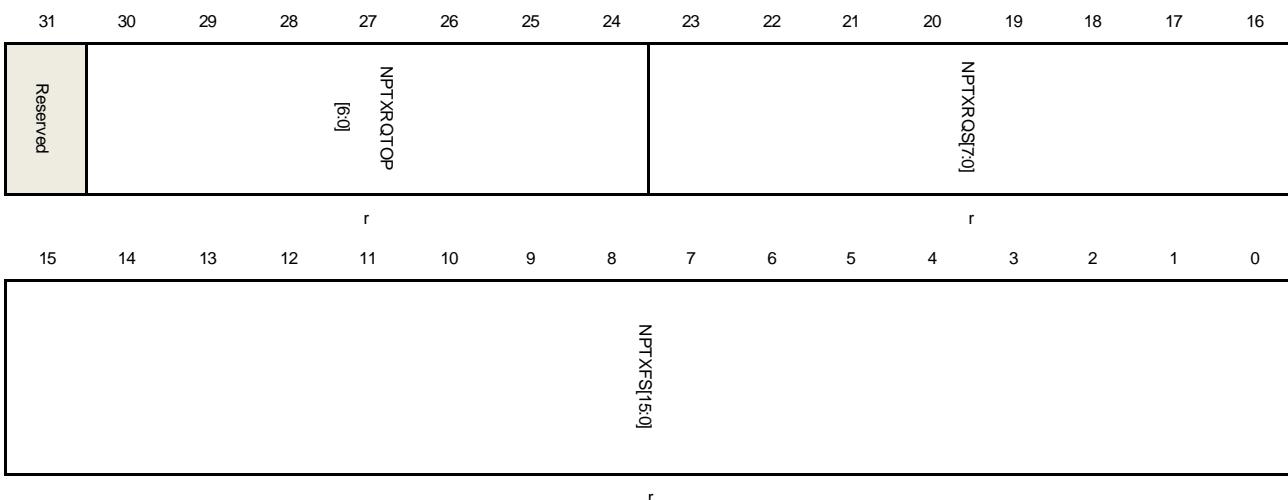
Address offset: 0x002C

Reset value: 0x0008 0200

This register reports the current status of the non-periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

Note: In Device mode, this register is not valid.

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value

30:24	NPTXRQTOP[6:0]	Top entry of the non-periodic Tx request queue Entry in the non-periodic transmit request queue. Bits 30:27: Channel number Bits 26:25: <ul style="list-style-type: none">- 00: IN/OUT token- 01: Zero-length OUT packet- 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	NPTXRQS[7:0]	Non-periodic Tx request queue space The remaining space of the non-periodic transmit request queue. 0: Request queue is Full 1: 1 entry 2: 2 entries ... n: n entries (0≤n≤8) Others: Reserved
15:0	NPTXFS[15:0]	Non-periodic Tx FIFO space The remaining space of the non-periodic transmit FIFO. In terms of 32-bit words. 0: Non-periodic Tx FIFO is full 1: 1 word 2: 2 words n: n words (0≤n≤NPTXFD) Others: Reserved

Global core configuration register (USBFS_GCCFG)

Address offset: 0x0038

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value
21	VBUSIG	<p>VBUS ignored</p> <p>When this bit is set, USBFS doesn't monitor the voltage on VBUS pin and always consider VBUS voltage as valid both in host mode and in device mode, then free the VBUS pin for other usage.</p> <p>0: VBUS is not ignored.</p> <p>1: VBUS is ignored and always consider VBUS voltage as valid.</p>
20	SOFOEN	<p>SOF output enable</p> <p>0: SOF pulse output disabled.</p> <p>1: SOF pulse output enabled.</p>
19	VBUSBCEN	<p>The VBUS B-device Comparer enable</p> <p>0: VBUS B-device comparer disabled</p> <p>1: VBUS B-device comparer enabled</p>
18	VBUSACEN	<p>The VBUS A-device Comparer enable</p> <p>0: VBUS A-device comparer disabled</p> <p>1: VBUS A-device comparer enabled</p>
17	Reserved	Must be kept at reset value
16	PWRON	<p>Power on</p> <p>This bit is the power switch for the internal embedded Full-Speed PHY.</p> <p>0: Embedded Full-Speed PHY power off.</p> <p>1: Embedded Full-Speed PHY power on.</p>
15:0	Reserved	Must be kept at reset value.

Core ID register (USBFS_CID)

Address offset: 0x003C

Reset value: 0x0000 1000

This register contains the Product ID.

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
CID[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0ID[15:0]

rw

Bits	Fields	Descriptions
31:0	CID[31:0]	Core ID Software can write or read this field and uses this field as a unique ID for its application

Host periodic transmit FIFO length register (USBFS_HPTFLEN)

Address offset: 0x0100

Reset value: 0x0200 0600

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
HPTXFD															
r/rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPTXFSA��															
r/rw															

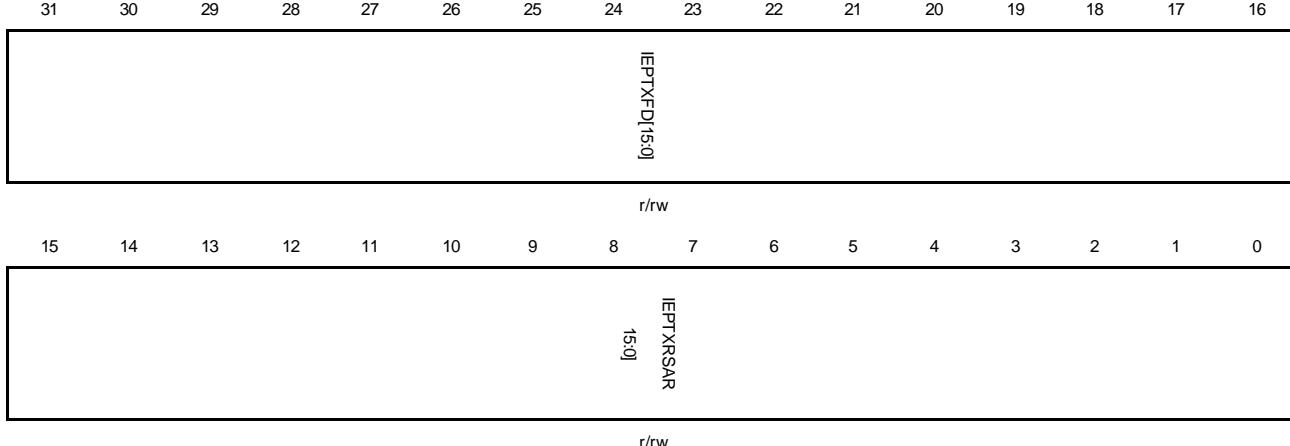
Bits	Fields	Descriptions
31:16	HPTXFD[15:0]	Host Periodic Tx FIFO depth In terms of 32-bit words. $1 \leq HPTXFD \leq 1024$
15:0	HPTXFSA��[15:0]	Host periodic Tx FIFO RAM start address The start address for host periodic transmit FIFO RAM is in term of 32-bit words.

Device IN endpoint transmit FIFO length register (USBFS_DIEPxTFLEN) (x = 1..3, where x is the FIFO_number)

Address offset: 0x0104 + (FIFO_number – 1) × 0x04

Reset value: 0x0200 0400

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



Bits Fields Descriptions

31:16	IEPTXFD[15:0]	IN endpoint Tx FIFO depth In terms of 32-bit words. $1 \leq HPTXFD \leq 1024$
15:0	IEPTXRSA R[15:0]	IN endpoint FIFO Tx RAM start address The start address for IN endpoint transmit FIFOx is in term of 32-bit words.

24.7.2. Host control and status registers

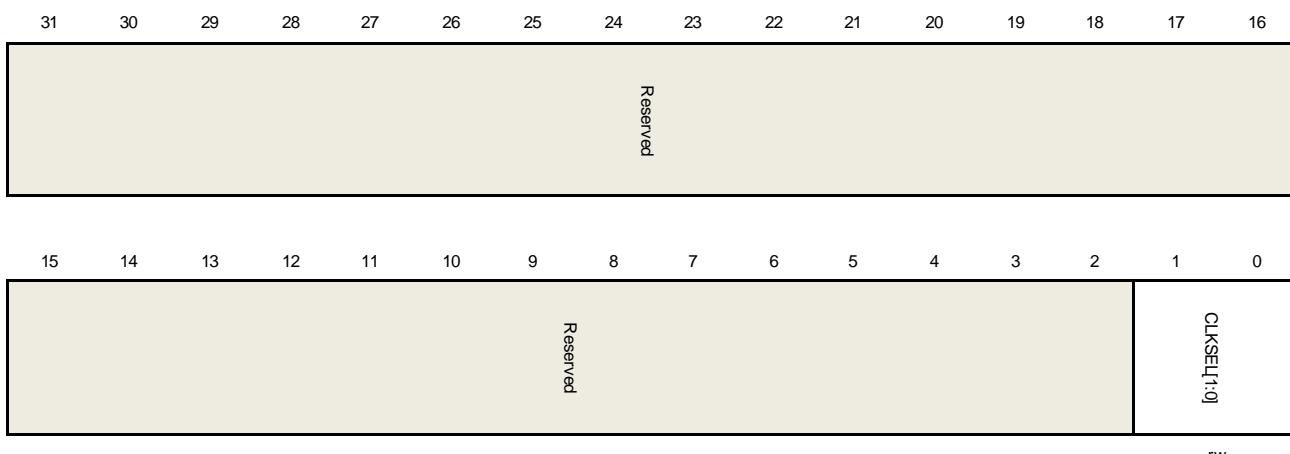
Host control register (USBFS_HCTL)

Address offset: 0x0400

Reset value: 0x0000 0000

This register configures the core after power on in host mode. Do not modify it after host initialization.

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1:0	CLKSEL[1:0]	Clock select for usbclock. 01: 48MHz clock others: reserved

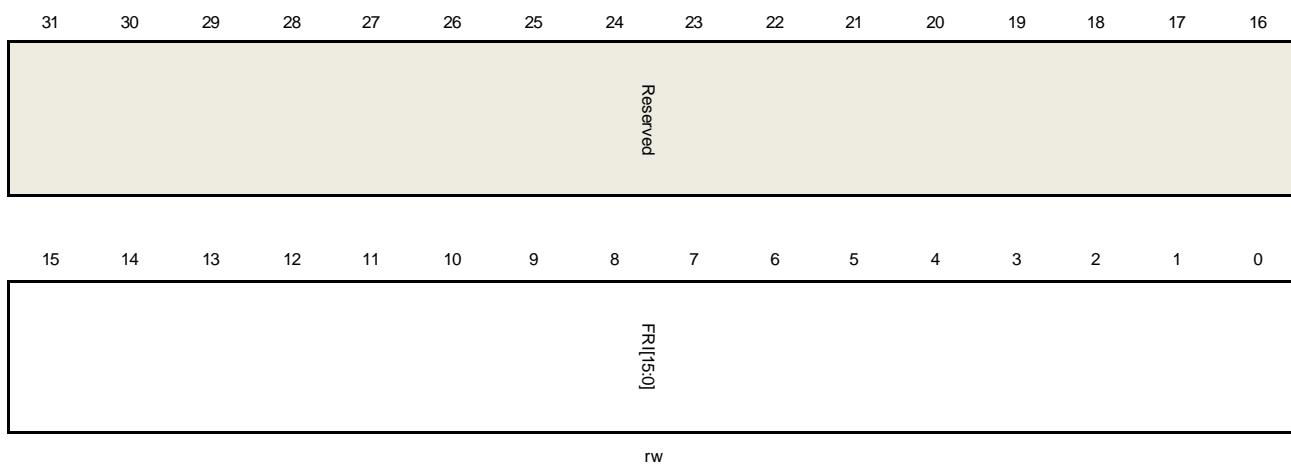
Host frame interval register (USBFS_HFT)

Address offset: 0x0404

Reset value: 0x0000 BB80

This register sets the frame interval for the current enumerating speed when USBFS controller is enumerating.

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



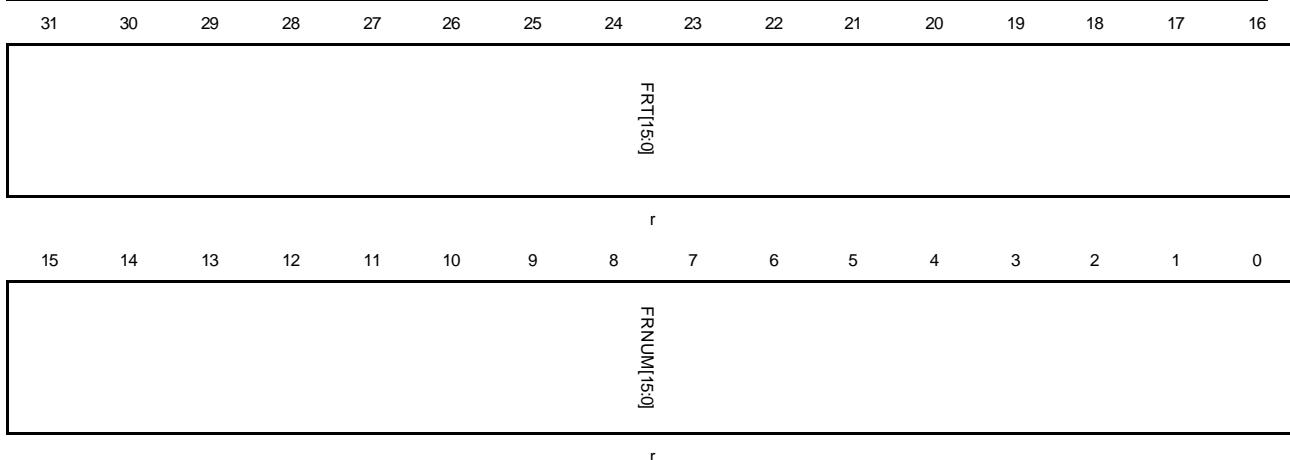
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	FRI[15:0]	Frame interval This value describes the frame time in terms of PHY clocks. Each time when port is enabled after a port reset operation, USBFS use a proper value according to the current speed, and software can write to this field to change the value. This value should be calculated using the frequency described below: Full-Speed: 48MHz Low-Speed: 6MHz

Host frame information remaining register (USBFS_HFINFR)

Address offset: 0x408

Reset value: 0xBB80 0000

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



Bits	Fields	Descriptions
31:16	FRT[15:0]	Frame remaining time This field reports the remaining time of current frame in terms of PHY clocks.
15:0	FRNUM[15:0]	Frame number This field reports the frame number of current frame and returns to 0 after it reaches 0x3FFF.

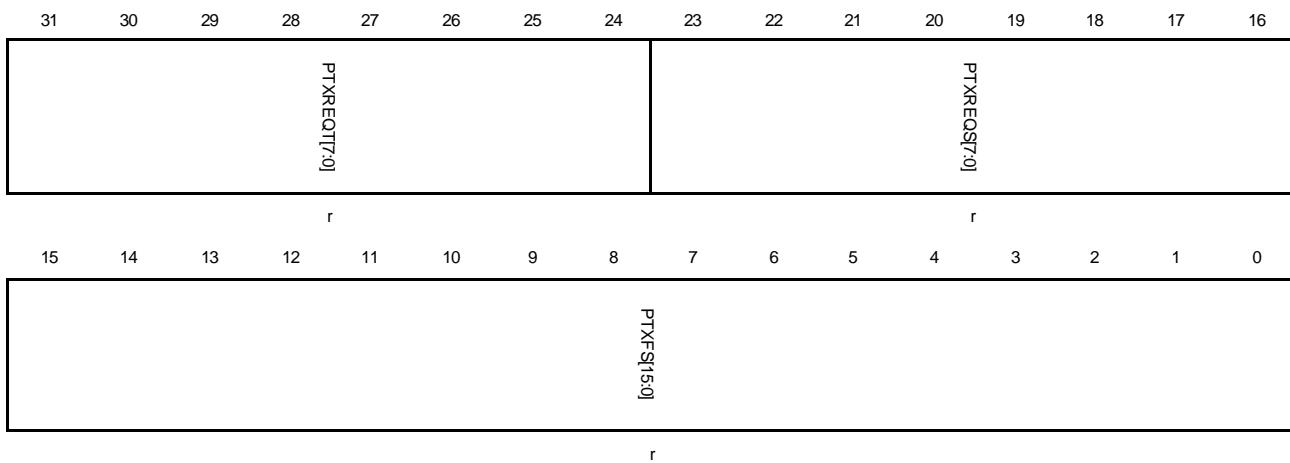
Host periodic transmit FIFO/queue status register (USBFS_HPTFQSTAT)

Address offset: 0x0410

Reset value: 0x0008 0200

This register reports the current status of the host periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

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



Bits	Fields	Descriptions
31:24	PTXREQT[7:0]	Top entry of the periodic Tx request queue

		Entry in the periodic transmit request queue.
		Bits 30:27: Channel Number
		Bits 26:25:
		00: IN/OUT token
		01: Zero-length OUT packet
		11: Channel halt request
		Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	PTXREQS[7:0]	<p>Periodic Tx request queue space</p> <p>The remaining space of the periodic transmit request queue.</p> <p>0: Request queue is Full</p> <p>1: 1 entry</p> <p>2: 2 entries</p> <p>...</p> <p>n: n entries ($0 \leq n \leq 8$)</p> <p>Others: Reserved</p>
15:0	PTXFS[15:0]	<p>Periodic Tx FIFO space</p> <p>The remaining space of the periodic transmit FIFO.</p> <p>In terms of 32-bit words.</p> <p>0: periodic Tx FIFO is full</p> <p>1: 1 word</p> <p>2: 2 words</p> <p>n: n words ($0 \leq n \leq \text{PTXFD}$)</p> <p>Others: Reserved</p>

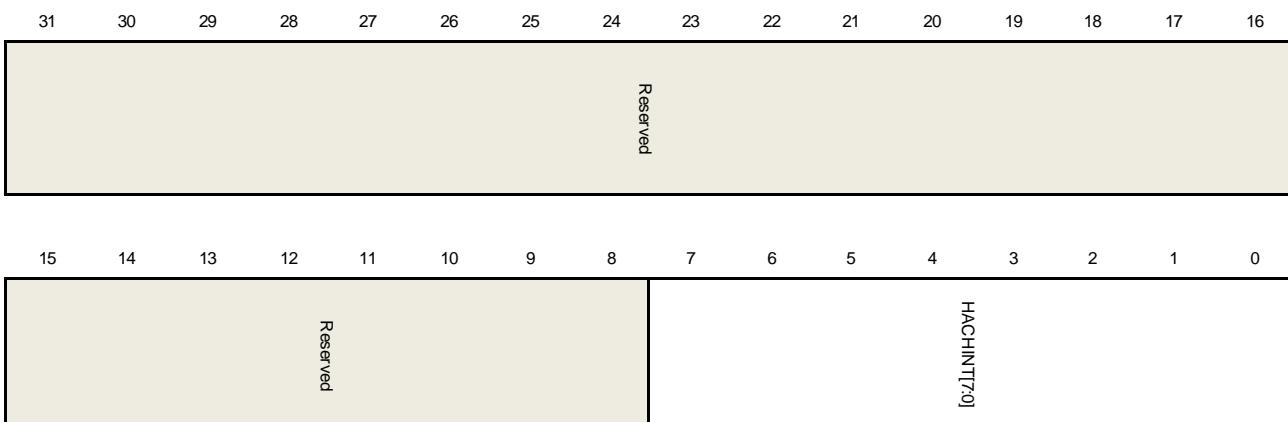
Host all channels interrupt register (USBFS_HACHINT)

Address offset: 0x0414

Reset value: 0x0000 0000

When a channel interrupt is triggered, USBFS set corresponding bit in this register and software should read this register to know which channel is asserting interrupts.

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	HACHINT[7:0]	Host all channel interrupts Each bit represents a channel: Bit 0 for channel 0, bit 7 for channel 7.

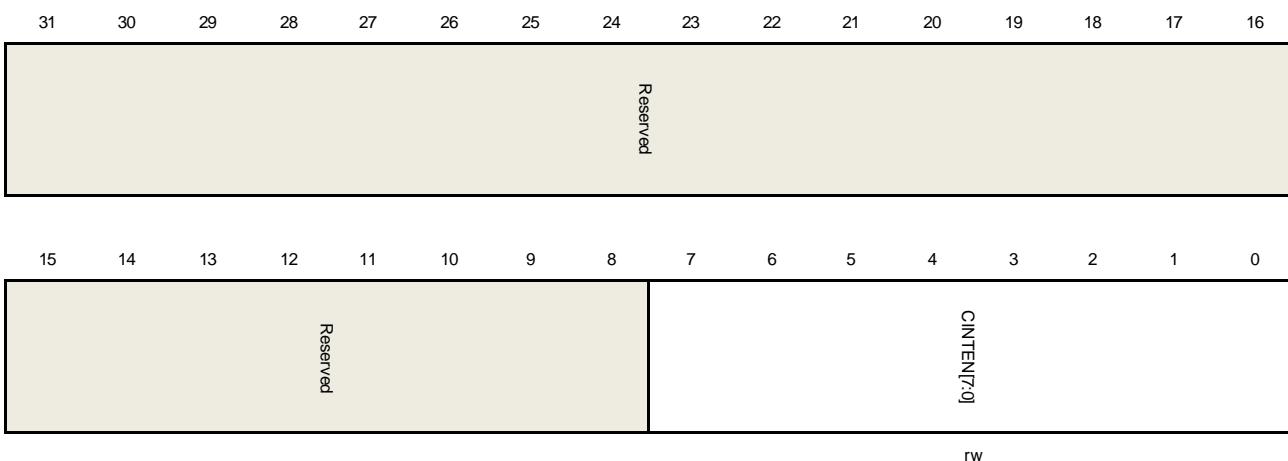
Host all channels interrupt enable register (USBFS_HACHINTEN)

Address offset: 0x0418

Reset value: 0x0000 0000

This register can be used by software to enable or disable a channel's interrupt. Only the channel whose corresponding bit in this register is set is able to cause the channel interrupt flag HCIF in USBFS_GINTF register.

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	CINTEN[7:0]	Channel interrupt enable 0: Disable channel-n interrupt 1: Enable channel-n interrupt Each bit represents a channel: Bit 0 for channel 0, bit 7 for channel 7.

Host port control and status register (USBFS_HPCS)

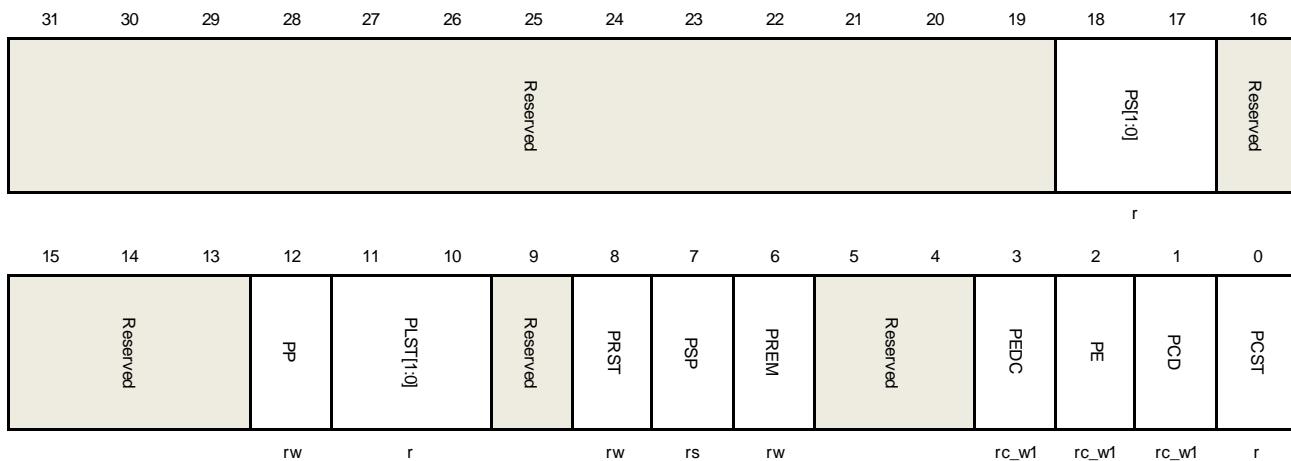
Address offset: 0x0440

Reset value: 0x0000 0000

This register controls the port's behavior and also has some flags which report the status of the port. The HPIF flag in USBFS_GINTF register will be triggered if one of these flags in this

register is set by USBFS: PRST, PEDC and PCD.

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



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18:17	PS[1:0]	Port speed Report the enumerated speed of the device attached to this port. 01: Full speed 10: Low speed Others: Reserved
16:13	Reserved	Must be kept at reset value
12	PP	Port power This bit should be set before a port is used. Because USBFS doesn't have power supply ability, it only uses this bit to know whether the port is in powered state. Software should ensure the true power supply on VBUS before setting this bit. 0: Port is powered off 1: Port is powered on
11:10	PLST[1:0]	Port line status Report the current state of USB data lines Bit 10: State of DP line Bit 11: State of DM line
9	Reserved	Must be kept at reset value
8	PRST	Port reset Application sets this bit to start a reset signal on USB port. Application should clear this bit when it wants to stop the reset signal. 0: Port is not in reset state 1: Port is in reset state

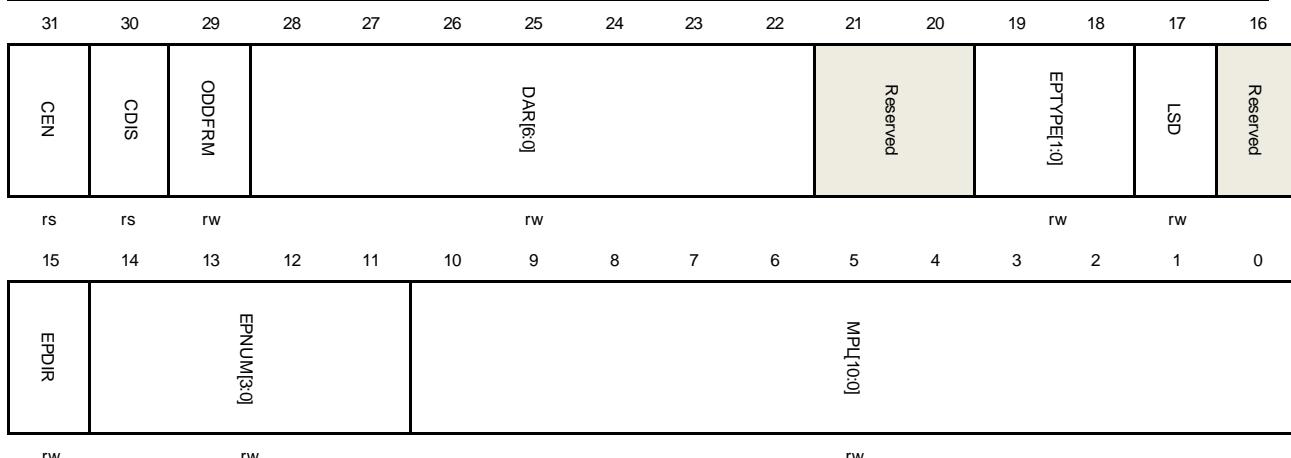
7	PSP	Port suspend
		Application sets this bit to put port into suspend state. When this bit is set the port stops sending SOF tokens. This bit can only be cleared by the following operations: PRST bit in this register is set by application PREM bit in this register is set A remote wakeup signal is detected A device disconnect is detected 0: Port is not in suspend state 1: Port is in suspend state
6	PREM	Port resume
		Application sets this bit to start a resume signal on USB port. Application should clear this bit when it wants to stop the resume signal. 0: No resume driven 1: Resume driven
5:4	Reserved	Must be kept at reset value
3	PEDC	Port enable/disable change Set by the core when the status of the Port enable bit 2 in this register changes.
2	PE	Port Enable This bit is automatically set by USBFS after a USB reset signal finishes and cannot be set by software. This bit is cleared by the following events: A disconnect condition Software clearing this bit 0: Port disabled 1: Port enabled
1	PCD	Port connect detected Set by USBFS when a device connection is detected. This bit can be cleared by writing 1 to this bit.
0	PCST	Port connect status 0: Device is not connected to the port 1: Device is connected to the port

Host channel-x control register (USBFS_HCHxCTL) (x = 0..7 where x = channel_number)

Address offset: 0x0500 + (channel_number × 0x20)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	CEN	<p>Channel enable</p> <p>Set by the application and cleared by USBFS.</p> <p>0: Channel disabled</p> <p>1: Channel enabled</p> <p>Software should follow the operation guide to disable or enable a channel.</p>
30	CDIS	<p>Channel disable</p> <p>Software can set this bit to disable the channel from processing transactions.</p> <p>Software should follow the operation guide to disable or enable a channel.</p>
29	ODDFRM	<p>Odd frame</p> <p>For periodic transfers (interrupt or isochronous transfer), this bit controls whether in an odd frame or even frame this channel's transaction is desired to be processed.</p> <p>0: Even frame</p> <p>1: Odd frame</p>
28:22	DAR[6:0]	<p>Device address</p> <p>The address of the USB device that this channel wants to communicate with.</p>
21:20	Reserved	Must be kept at reset value
19:18	EPTYPE[1:0]	<p>Endpoint type</p> <p>The transfer type of the endpoint that this channel wants to communicate with.</p> <p>00: Control</p> <p>01: Isochronous</p> <p>10: Bulk</p> <p>11: Interrupt</p>
17	LSD	<p>Low-Speed device</p> <p>The device that this channel wants to communicate with is a Low-Speed Device.</p>
16	Reserved	Must be kept at reset value

15	EPDIR	Endpoint direction The transfer direction of the endpoint that this channel wants to communicate with. 0: OUT 1: IN
14:11	EPNUM[3:0]	Endpoint number The number of the endpoint that this channel wants to communicate with.
10:0	MPL[10:0]	Maximum packet length The target endpoint's maximum packet length.

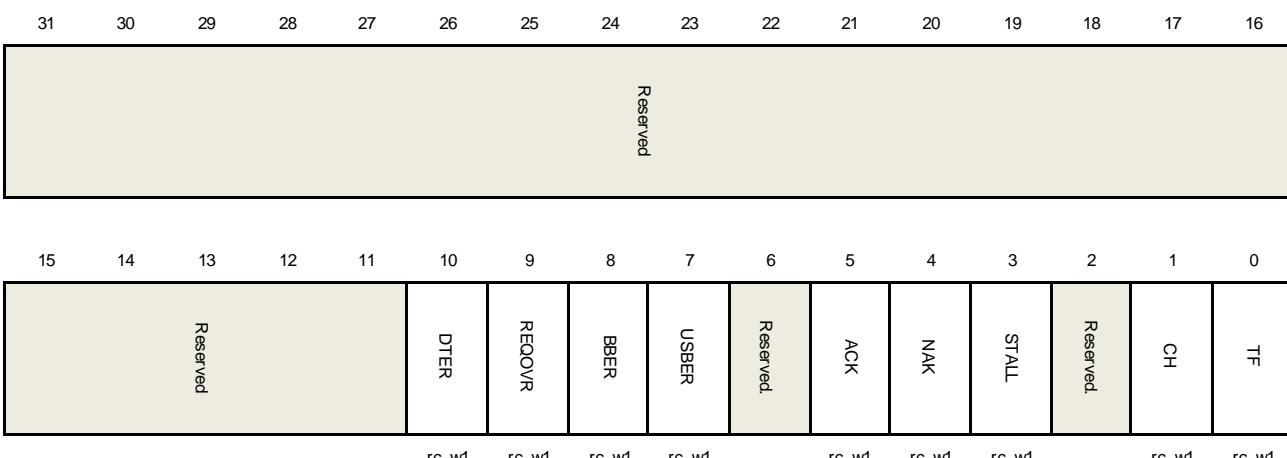
Host channel-x interrupt flag register (USBFS_HCHxINTF) (x = 0..7 where x = channel number)

Address offset: 0x0508 + (channel_number × 0x20)

Reset value: 0x0000 0000

This register contains the status and events of a channel, when software get a channel interrupt, it should read this register for the respective channel to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

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



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
10	DTER	Data toggle error The IN transaction gets a data packet but the PID of this packet doesn't match DPID [1:0] bits in USBFS_HCHxLEN register.
9	REQOVR	Request queue overrun The periodic request queue is full when software starts new transfers.
8	BBER	Babble error A babble condition occurs on USB bus. A typical reason for babble condition is that

a device sends a data packet and the packet length exceeds the endpoint's maximum packet length.

7	USBER	USB Bus Error
		The USB error flag is set when the following conditions occurs during receiving a packet:
		A received packet has a wrong CRC field
		A stuff error detected on USB bus
		Timeout when waiting for a response packet
6	Reserved	Must be kept at reset value
5	ACK	ACK
		An ACK response is received or transmitted
4	NAK	NAK
		A NAK response is received.
3	STALL	STALL
		A STALL response is received.
2	Reserved	Must be kept at reset value
1	CH	Channel halted
		This channel is disabled by a request, and it will not response to other requests during the request processing.
0	TF	Transfer finished
		All the transactions of this channel finish successfully, and no error occurs. For IN channel, this flag will be triggered after PCNT bits in USBFS_HCHxLEN register reach zero. For OUT channel, this flag will be triggered when software reads and pops a TF status entry from the RxFIFO.

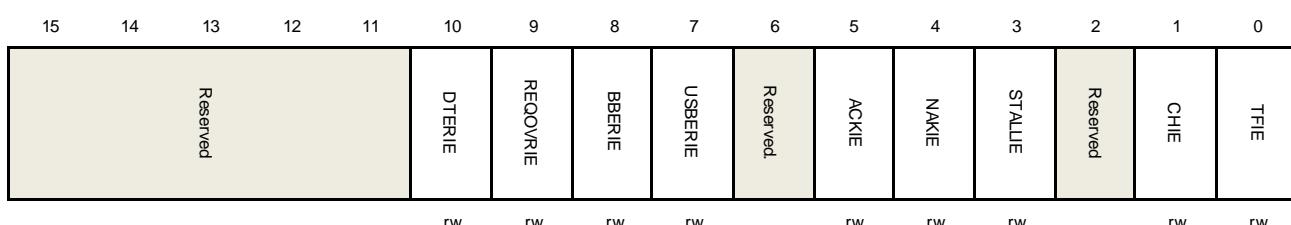
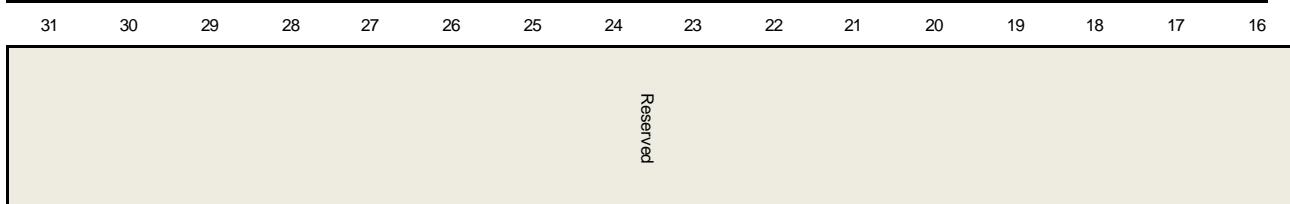
Host channel-x interrupt enable register (USBFS_HCHxINTEN) (x = 0..7, where x = channel number)

Address offset: 0x050C + (channel_number × 0x20)

Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_HCHxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_HCHxINTF register is able to trigger a channel interrupt. The bits in this register are set and cleared by software.

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



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
10	DTERIE	Data toggle error interrupt enable 0: Disable data toggle error interrupt 1: Enable data toggle error interrupt
9	REQOVRIE	Request queue overrun interrupt enable 0: Disable request queue overrun interrupt 1: Enable request queue overrun interrupt
8	BBERIE	Babble error interrupt enable 0: Disable babble error interrupt 1: Enable babble error interrupt
7	USBERIE	USB bus error interrupt enable 0: Disable USB bus error interrupt 1: Enable USB bus error interrupt
6	Reserved	Must be kept at reset value
5	ACKIE	ACK interrupt enable 0: Disable ACK interrupt 1: Enable ACK interrupt
4	NAKIE	NAK interrupt enable 0: Disable NAK interrupt 1: Enable NAK interrupt
3	STALLIE	STALL interrupt enable 0: Disable STALL interrupt 1: Enable STALL interrupt
2	Reserved	Must be kept at reset value

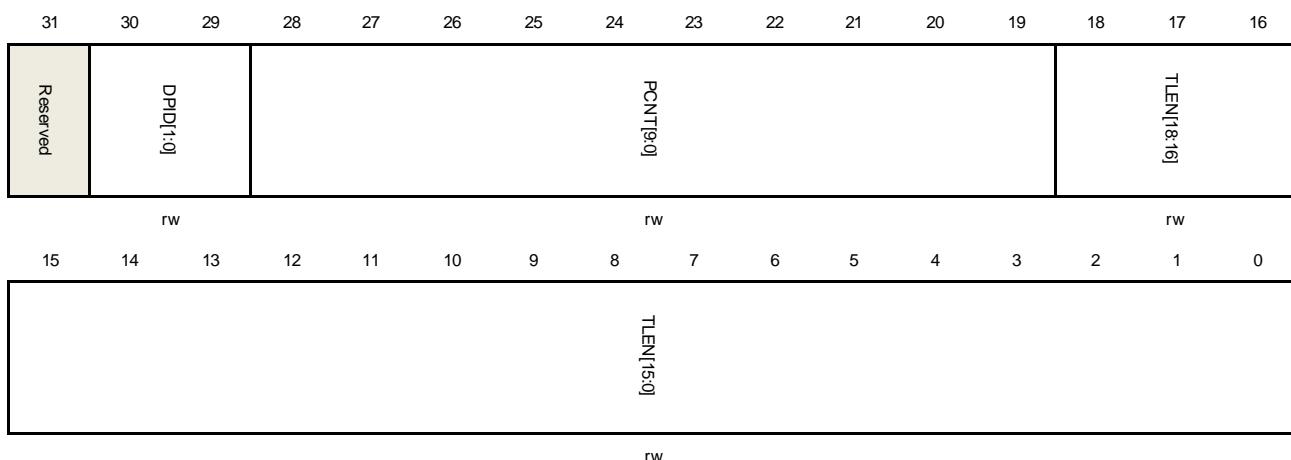
1	CHIE	Channel halted interrupt enable 0: Disable channel halted interrupt 1: Enable channel halted interrupt
0	TFIE	Transfer finished interrupt enable 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

Host channel-x transfer length register (USBFS_HCHxLEN) (x = 0..7, where x = channel number)

Address offset: 0x0510 + (channel_number × 0x20)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:29	DPID[1:0]	Data PID Software should write this field before the transfer starts. For OUT transfers, this field controls the Data PID of the first transmitted packet. For IN transfers, this field controls the expected Data PID of the first received packet, and DTERR will be triggered if the Data PID doesn't match. After the transfer starts, USBFS changes and toggles this field automatically following the USB protocol. 00: DATA0 10: DATA1 11: SETUP (For control transfer only) 01: Reserved
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted (OUT) or received (IN) in a transfer.

Software should program this field before the channel is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.

18:0	TLEN[18:0]	<p>Transfer length</p> <p>The total data bytes number of a transfer.</p> <p>For OUT transfers, this field is the total data bytes of all the data packets desired to be transmitted in an OUT transfer. Software should program this field before the channel is enabled. When software successfully writes a packet into the channel's data TxFIFO, this field is decreased by the byte size of the packet.</p> <p>For IN transfer each time software or DMA reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.</p>
------	------------	---

24.7.3. Device control and status registers

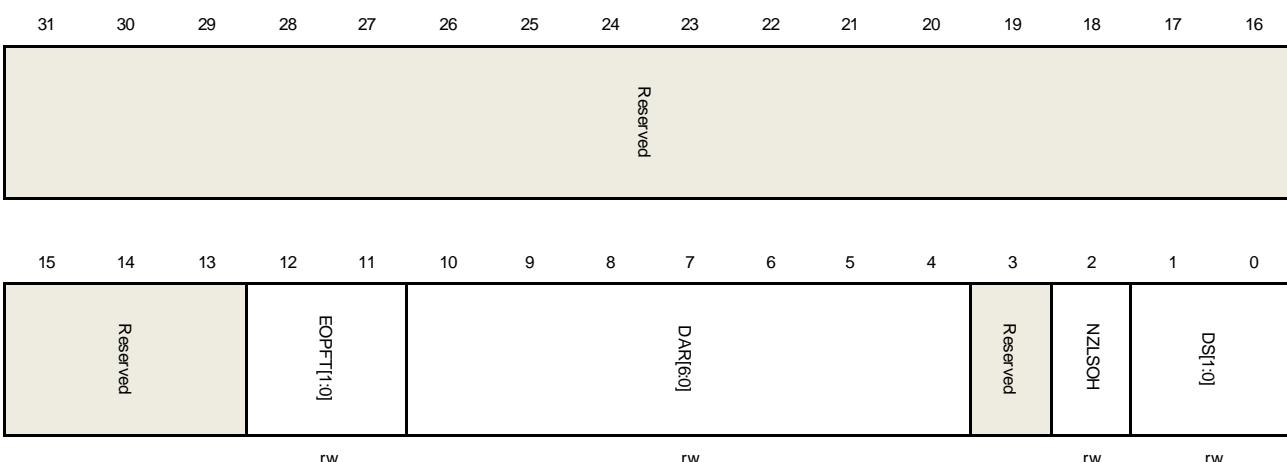
Device configuration register (USBFS_DCFG)

Address offset: 0x0800

Reset value: 0x0000 0000

This register configures the core in device mode after power on or after certain control commands or enumeration. Do not change this register after device initialization.

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



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value
12:11	EOPFT[1:0]	<p>End of periodic frame time</p> <p>This field defines the percentage time point in a frame that the end of periodic frame (EOPF) flag should be triggered.</p> <p>00: 80% of the frame time</p> <p>01: 85% of the frame time</p>

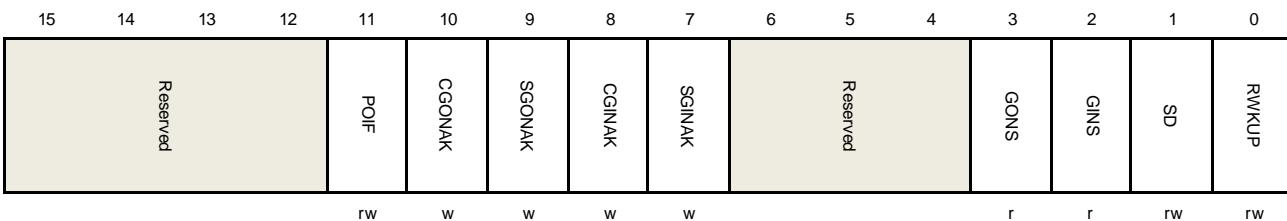
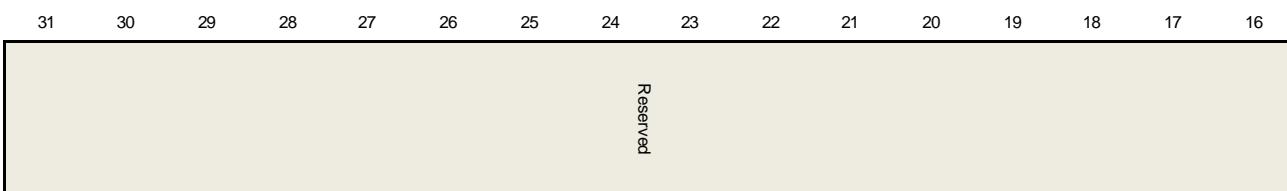
		10: 90% of the frame time 11: 95% of the frame time
10:4	DAR[6:0]	Device address This field defines the USB device's address. USBFS uses this field to match with the incoming token's device address field. Software should program this field after receiving a Set Address command from USB host.
3	Reserved	Must be kept at reset value
2	NZLSOH	Non-zero-length status OUT handshake When a USB device receives a non-zero-length data packet during status OUT stage, this field controls that either USBFS should receive this packet or reject this packet with a STALL handshake. 0: Treat this packet as a normal packet and response according to the status of NAKS and STALL bits in USBFS_DOEPxCTL register. 1: Send a STALL handshake and don't save the received OUT packet.
1:0	DS[1:0]	Device speed This field controls the device speed when the device connected to a host. 11: Full speed Others: Reserved

Device control register (USBFS_DCTL)

Address offset: 0x0804

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	POIF	Power-on initialization finished

		Software should set this bit to notify USBFS that the registers are initialized after waking up from power down state.
10	CGONAK	<p>Clear global OUT NAK</p> <p>Software sets this bit to clear GONS bit in this register.</p>
9	SGONAK	<p>Set global OUT NAK</p> <p>Software sets this bit to set GONS bit in this register.</p> <p>When GONS bit is zero, setting this bit will also cause GONAK flag in USBFS_GINTF register triggered after a while. Software should clear the GONAK flag before writing this bit again.</p>
8	CGINAK	<p>Clear global IN NAK</p> <p>Software sets this bit to clear GINS bit in this register.</p>
7	SGINAK	<p>Set global IN NAK</p> <p>Software sets this bit to set GINS bit in this register.</p> <p>When GINS bit is zero, setting this bit will also cause GINAK flag in USBFS_GINTF register triggered after a while. Software should clear the GINAK flag before writing this bit again.</p>
6:4	Reserved	Must be kept at reset value
3	GONS	<p>Global OUT NAK status</p> <p>0: The handshake that USBFS response to OUT transaction packet and whether to save the OUT data packet are decided by Rx FIFO status, endpoint's NAK and STALL bits.</p> <p>1: USHBS always responds to OUT transaction with NAK handshake and doesn't save the incoming OUT data packet.</p>
2	GINS	<p>Global IN NAK status</p> <p>0: The response to IN transaction is decided by Tx FIFO status, endpoint's NAK and STALL bits.</p> <p>1: USBFS always responds to IN transaction with a NAK handshake.</p>
1	SD	<p>Soft disconnect</p> <p>Software can use this bit to generate a soft disconnect condition on USB bus. After this bit is set, USBFS switches off the pull up resistor on DP line. This will cause the host to detect a device disconnect.</p> <p>0: No soft disconnect generated.</p> <p>1: Generate a soft disconnection.</p>
0	RWKUP	<p>Remote wakeup</p> <p>In suspend state, software can use this bit to generate a Remote wake up signal to inform host that it should resume the USB bus.</p> <p>0: No remote wakeup signal generated.</p> <p>1: Generate remote wakeup signal.</p>

Device status register (USBFS_DSTAT)

Address offset: 0x0808

Reset value: 0x0000 0000

This register contains status and information of the USBFS in device mode.

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value
21:8	FNRSOF[13:0]	The frame number of the received SOF. USBFS always update this field after receiving a SOF token
7:3	Reserved	Must be kept at reset value
2:1	ES[1:0]	Enumerated speed This field reports the enumerated device speed. Read this field after the ENUMF flag in USBFS_GINTF register is triggered. 11: Full speed Others: reserved
0	SPST	Suspend status This bit reports whether device is in suspend state. 0: Device is not in suspend state. 1: Device is in suspend state.

Device IN endpoint common interrupt enable register (USBFS_DIEPINTEN)

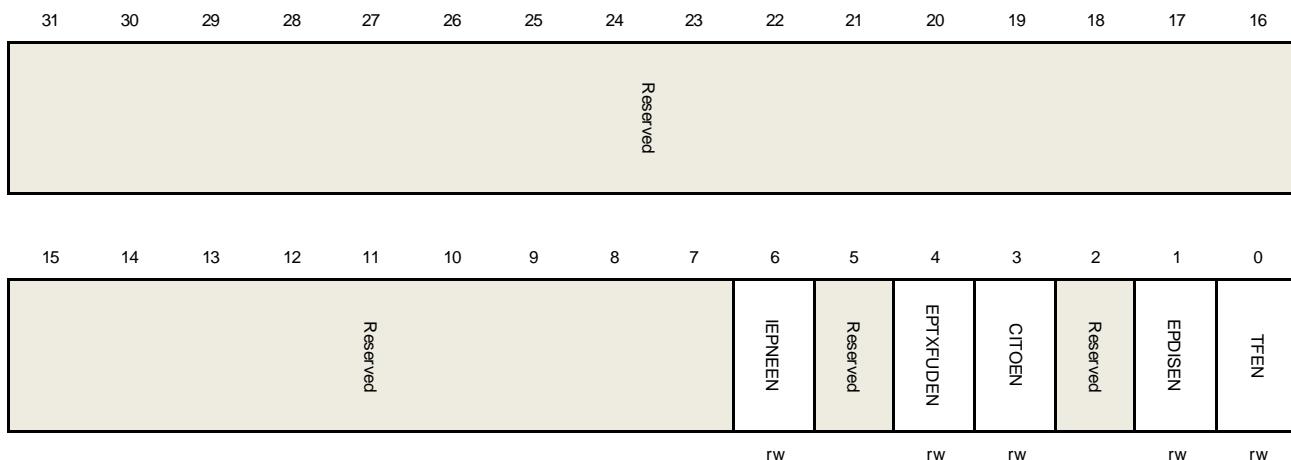
Address offset: 0x810

Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_DIEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_DIEPxINTF register

is able to trigger an endpoint interrupt in USBFS_DAEPIINT register. The bits in this register are set and cleared by software.

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	IEPNEEN	IN endpoint NAK effective interrupt enable bit 0: Disable IN endpoint NAK effective interrupt 1: Enable IN endpoint NAK effective interrupt
5	Reserved	Must be kept at reset value
4	EPTXFUDEN	Endpoint Tx FIFO underrun interrupt enable bit 0: Disable endpoint Tx FIFO underrun interrupt 1: Enable endpoint Tx FIFO underrun interrupt
3	CITOEN	Control In timeout interrupt enable bit 0: Disable control In timeout interrupt 1: Enable control In timeout interrupt
2	Reserved	Must be kept at reset value
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable endpoint disabled interrupt 1: Enable endpoint disabled interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

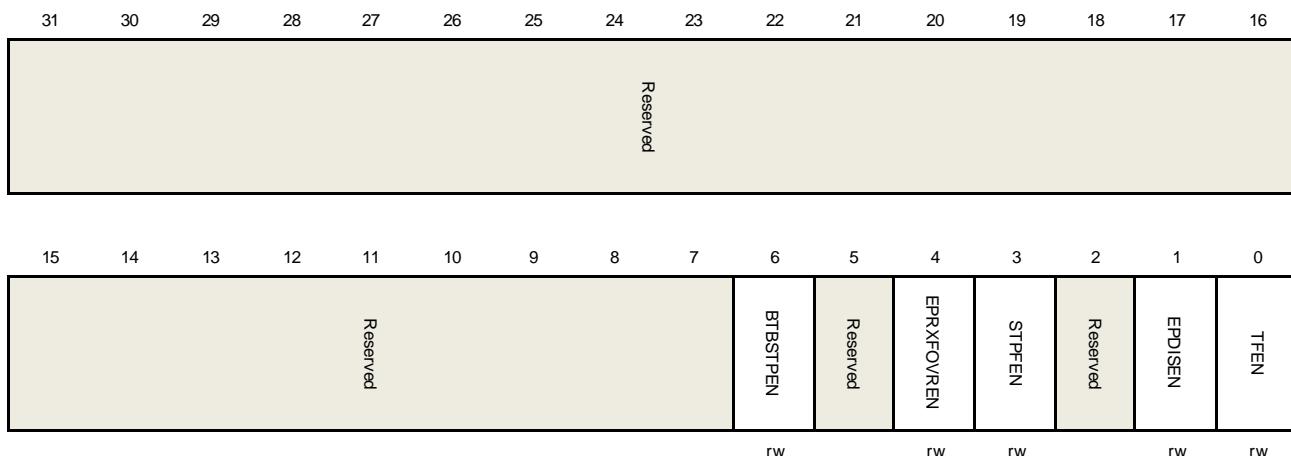
Device OUT endpoint common interrupt enable register (USBFS_DOEPINTEN)

Address offset: 0x0814

Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_DOEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_DOEPxINTF register is able to trigger an endpoint interrupt in USBFS_DAEPINT register. The bits in this register are set and cleared by software.

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	BTBSPEN	Back-to-back SETUP packets (Only for control OUT endpoint) interrupt enable bit 0: Disable back-to-back SETUP packets interrupt 1: Enable back-to-back SETUP packets interrupt
5	Reserved	Must be kept at reset value
4	EPRXFOVREN	Endpoint Rx FIFO overrun interrupt enable bit 0: Disable endpoint Rx FIFO overrun interrupt 1: Enable endpoint Rx FIFO overrun interrupt
3	STPFEN	SETUP phase finished (Only for control OUT endpoint) interrupt enable bit 0: Disable SETUP phase finished interrupt 1: Enable SETUP phase finished interrupt
2	Reserved	Must be kept at reset value
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable endpoint disabled interrupt 1: Enable endpoint disabled interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

Device all endpoints interrupt register (USBFS_DAEPINT)

Address offset: 0x0818

Reset value: 0x0000 0000

When an endpoint interrupt is triggered, USBFS sets corresponding bit in this register and software should read this register to know which endpoint is asserting an interrupt.

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19:16	OEPITB[3:0]	Device all OUT endpoint interrupt bits Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 19 for OUT endpoint 3.
15:4	Reserved	Must be kept at reset value
3:0	IEPITB[3:0]	Device all IN endpoint interrupt bits Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3.

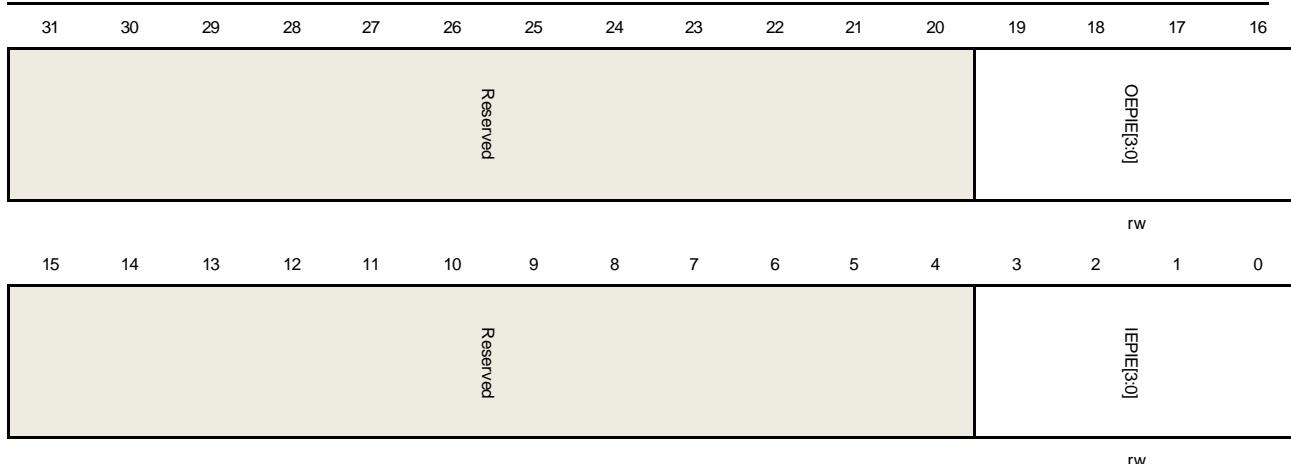
Device all endpoints interrupt enable register (USBFS_DAEPINTEN)

Address offset: 0x081C

Reset value: 0x0000 0000

This register can be used by software to enable or disable an endpoint's interrupt. Only the endpoint whose corresponding bit in this register is set is able to cause the endpoint interrupt flag OEPIF or IEPIF in USBFS_GINTF register.

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



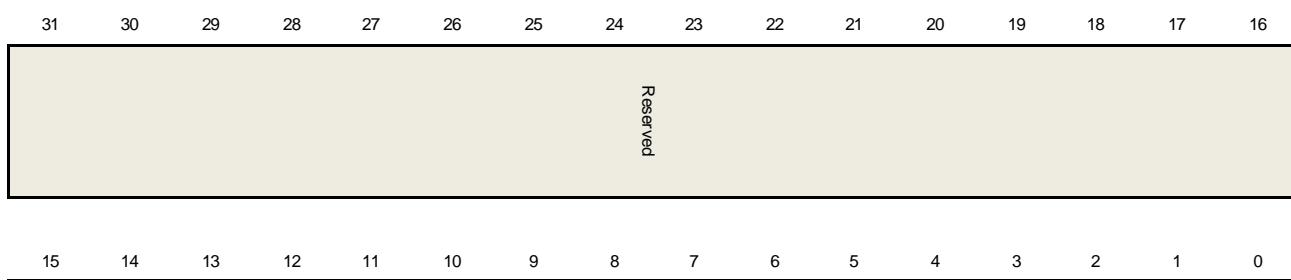
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19:16	OPIE[3:0]	Out endpoint interrupt enable 0: Disable OUT endpoint-n interrupt 1: Enable OUT endpoint-n interrupt Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 19 for OUT endpoint 3.
15:4	Reserved	Must be kept at reset value
3:0	IEPIE[3:0]	IN endpoint interrupt enable bits 0: Disable IN endpoint-n interrupt 1: Enable IN endpoint-n interrupt Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3.

Device VBUS discharge time register (USBFS_DVBUSDT)

Address offset: 0x0828

Reset value: 0x0000 17D7

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





rw

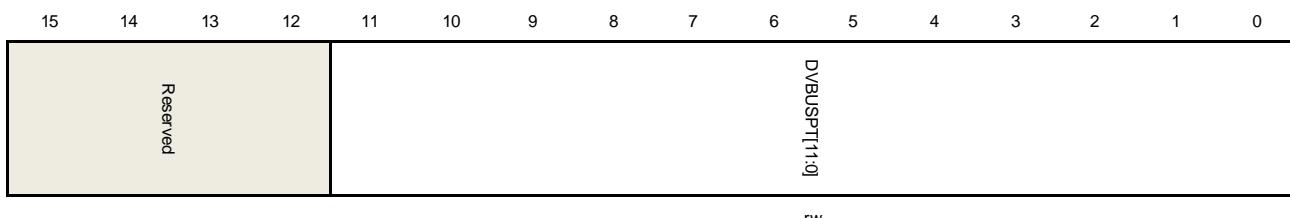
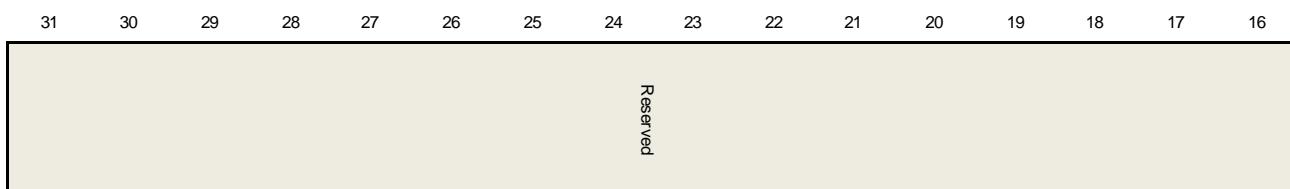
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	DVBUSDT[15:0]	Device V _{BUS} discharge time There is a discharge process after V _{BUS} pulsing in SRP protocol. This field defines the discharge time of V _{BUS} . The true discharge time is 1024 * DVBUSDT[15:0] *TUSBCLOCK, where TUSBCLOCK is the period time of USB clock.

Device VBUS pulsing time register (USBFS_DVBUSPT)

Address offset: 0x082C

Reset value: 0x0000 05B8

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



rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	DVBUSPT[11:0]	Device V _{BUS} pulsing time This field defines the pulsing time for V _{BUS} . The true pulsing time is 1024*DVBUSPT[11:0] *TUSBCLOCK, where TUSBCLOCK is the period time of USB clock.

Device IN endpoint FIFO empty interrupt enable register (USBFS_DIEPFEINTEN)

Address offset: 0x0834

Reset value: 0x0000 0000

This register contains the enable bits for the Tx FIFO empty interrupts of IN endpoints.

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



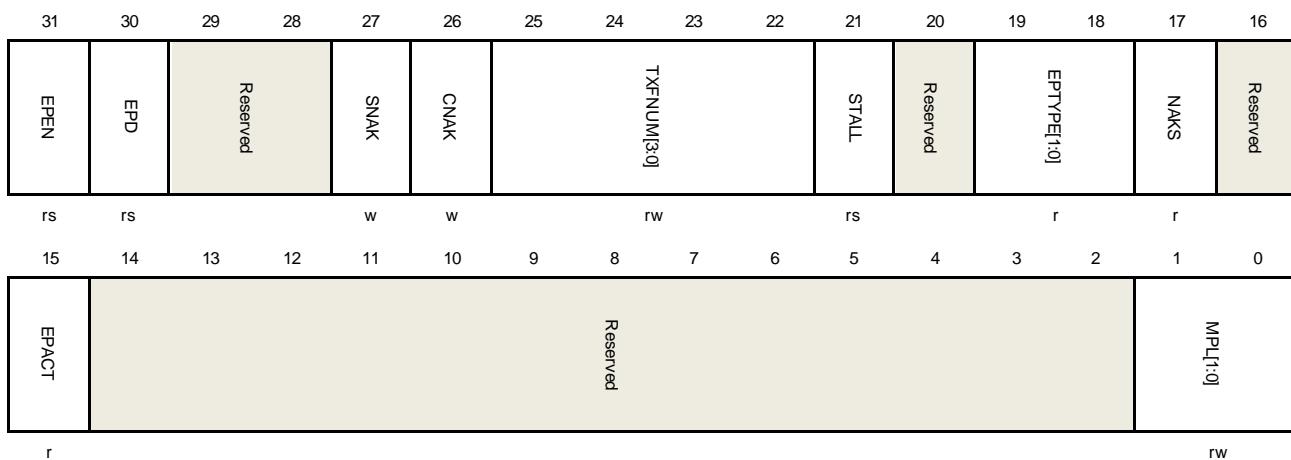
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	IEPTXFEIE[3:0]	<p>IN endpoint Tx FIFO empty interrupt enable bits</p> <p>This field controls whether the TXFE bits in USBFS_DIEPxINTF registers are able to generate an endpoint interrupt bit in USBFS_DAEPINT register.</p> <p>Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3</p> <p>0: Disable FIFO empty interrupt</p> <p>1: Enable FIFO empty interrupt</p>

Device IN endpoint 0 control register (USBFS_DIEP0CTL)

Address offset: 0x0900

Reset value: 0x0000 8000

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



Bits	Fields	Descriptions
31	EPEN	<p>Endpoint enable</p> <p>Set by the application and cleared by USBFS.</p> <p>0: Endpoint disabled</p> <p>1: Endpoint enabled</p> <p>Software should follow the operation guide to disable or enable an endpoint.</p>
30	EPD	<p>Endpoint disable</p> <p>Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.</p>
29:28	Reserved	Must be kept at reset value
27	SNAK	<p>Set NAK</p> <p>Software sets this bit to set NAKS bit in this register.</p>
26	CNAK	<p>Clear NAK</p> <p>Software sets this bit to clear NAKS bit in this register.</p>
25:22	TXFNUM[3:0]	<p>Tx FIFO number</p> <p>Defines the Tx FIFO number of IN endpoint 0.</p>
21	STALL	<p>STALL handshake</p> <p>Software can set this bit to make USBFS sends STALL handshake when receiving IN token. USBFS will clear this bit after a SETUP token is received on the corresponding OUT endpoint 0. This bit has a higher priority than NAKS bit in this register and GINS bit in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect.</p>
20	Reserved	Must be kept at reset value
19:18	EPTYPE[1:0]	<p>Endpoint type</p> <p>This field is fixed to '00' for control endpoint.</p>
17	NAKS	<p>NAK status</p> <p>This bit controls the NAK status of USBFS when both STALL bit in this register and GINS bit in USBFS_DCTL register are cleared:</p> <p>0: USBFS sends data or handshake packets according to the status of the endpoint's Tx FIFO.</p> <p>1: USBFS always sends NAK handshake to the IN token.</p> <p>This bit is read-only and software should use CNAK and SNAK in this register to control this bit.</p>
16	Reserved	Must be kept at reset value
15	EPACT	<p>Endpoint active</p> <p>This field is fixed to '1' for endpoint 0.</p>
14:2	Reserved	Must be kept at reset value

1:0	MPL[1:0]	Maximum packet length This field defines the maximum packet length for a control data packet. As described in USB 2.0 protocol, there are 4 kinds of length for control transfers: 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes
-----	----------	--

Device IN endpoint-x control register (USBFS_DIEPxCTL) (x = 1..3, where x = endpoint_number)

Address offset: 0x0900 + (endpoint_number × 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
EPEN	EPD	SODDFRM/SD1 PID	SD0PID/SEVNF RM	SNAK	CNAK		TXFNUM[3:0]			STALL	Reserved	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w		rw			rw/rs		rw	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	Reserved									MP4[10:0]					
rw										rw					

Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous IN endpoints) This bit has effect only if this is an isochronous IN endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk IN endpoints) Software sets this bit to set DPID bit in this register.

28	SEVENFRM	Set even frame (For isochronous IN endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk IN endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	TXFNUM[3:0]	Tx FIFO number Defines the Tx FIFO number of this IN endpoint.
21	STALL	STALL handshake Software can set this bit to make USBFS sends STALL handshake when receiving IN token. This bit has a higher priority than NAKS bit in this register and GINS bit in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect. For control IN endpoint: Only USBFS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it. For interrupt or bulk IN endpoint: Only software can clear this bit
20	Reserved	Must be kept at reset value
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this endpoint: 00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	NAKS	NAK status This bit controls the NAK status of USBFS when both STALL bit in this register and GINS bit in USBFS_DCTL register are cleared: 0: USBFS sends data or handshake packets according to the status of the endpoint's Tx FIFO. 1: USBFS always sends NAK handshake to the IN token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	EOFMR	Even/odd frame (For isochronous IN endpoints) For isochronous transfers, software can use this bit to control that USBFS only sends data packets for IN tokens in even or odd frames. If the parity of the current frame number doesn't match with this bit, USBFS only responses with a zero-length packet.

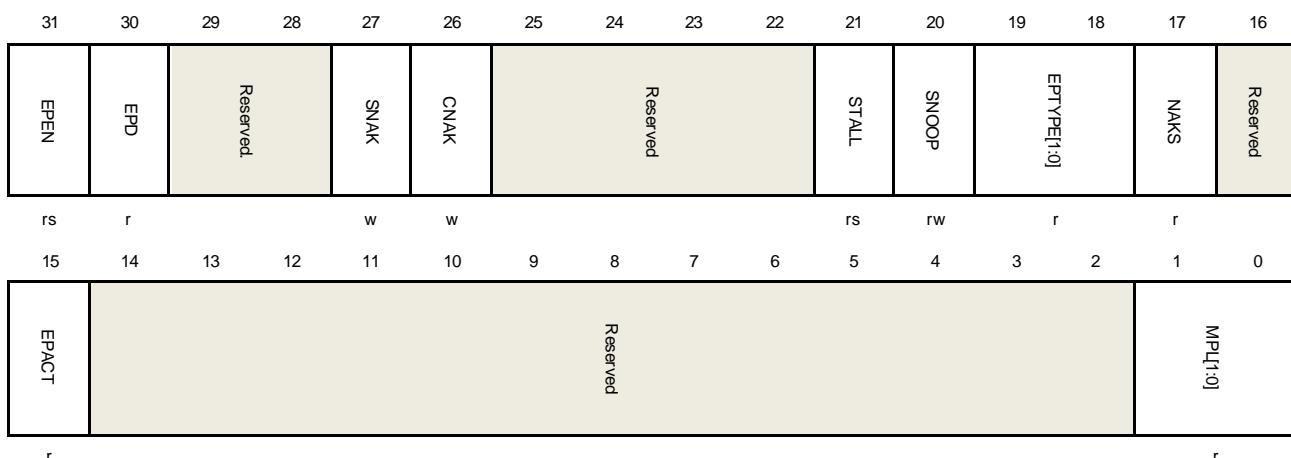
		0: Only sends data in even frames 1: Only sends data in odd frames
	DPID	Endpoint data PID (For interrupt/bulk IN endpoints) There is a data PID toggle scheme in interrupt or bulk transfer. Set SD0PID to set this bit before a transfer starts and USBFS maintains this bit during transfers according to the data toggle scheme described in USB protocol.
		0: Data packet's PID is DATA0 1: Data packet's PID is DATA1
15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

Device OUT endpoint 0 control register (USBFS_DOEP0CTL)

Address offset: 0x0B00

Reset value: 0x0000 8000

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



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable This bit is fixed to 0 for OUT endpoint 0.

29:28	Reserved	Must be kept at reset value
27	SNAK	<p>Set NAK</p> <p>Software sets this bit to set NAKS bit in this register.</p>
26	CNAK	<p>Clear NAK</p> <p>Software sets this bit to clear NAKS bit in this register</p>
25:22	Reserved	Must be kept at reset value
21	STALL	<p>STALL handshake</p> <p>Set this bit to make USBFS send STALL handshake during an OUT transaction. USBFS will clear this bit after a SETUP token is received on OUT endpoint 0. This bit has a higher priority than NAKS bit in this register, i.e. if both STALL and NAKS bits are set, the STALL bit takes effect.</p>
20	SNOOP	<p>Snoop mode</p> <p>This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBFS doesn't check the received data packet's CRC value.</p> <p>0:Snoop mode disabled 1:Snoop mode enabled</p>
19:18	EPTYPE[1:0]	<p>Endpoint type</p> <p>This field is fixed to '00' for control endpoint.</p>
17	NAKS	<p>NAK status</p> <p>This bit controls the NAK status of USBFS when both STALL bit in this register and GONS bit in USBFS_DCTL register are cleared:</p> <p>0: USBFS sends data or handshake packets according to the status of the endpoint's Rx FIFO. 1: USBFS always sends NAK handshake for the OUT token.</p> <p>This bit is read-only and software should use CNAK and SNAK in this register to control this bit.</p>
16	Reserved	Must be kept at reset value
15	EPACT	<p>Endpoint active</p> <p>This field is fixed to '1' for endpoint 0.</p>
14:2	Reserved	Must be kept at reset value
1:0	MPL[1:0]	<p>Maximum packet length</p> <p>This is a read-only field, and its value comes from the MPL field of USBFS_DIEP0CTL register:</p> <p>00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes</p>

Device OUT endpoint-x control register (USBFS_DOEPxCTL) (x = 1..3, where x = endpoint_number)

Address offset: 0x0B00 + (endpoint_number × 0x20)

Reset value: 0x0000 0000

The application uses this register to control the operations of each logical OUT endpoint other than OUT endpoint 0.

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
EPEN	EPD	SODDFRM/SD1PID	SEVNFRM/SD0PID	SNAK	CNAK	Reserved				STALL	SNOOP	EPTYPE[1:0]	NAKS	EOFfrm/DPID	
rs	rs	w	w	w	w					rw/rs	rw	rw	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	Reserved				MPU100]										

Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous OUT endpoints) This bit has effect only if this is an isochronous OUT endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk OUT endpoints) Software sets this bit to set DPID bit in this register.
28	SEVNFRM	Set even frame (For isochronous OUT endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk OUT endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK

		Software sets this bit to set NAKS bit in this register.
26	CNAK	<p>Clear NAK</p> <p>Software sets this bit to clear NAKS bit in this register.</p>
25:22	Reserved	Must be kept at reset value
21	STALL	<p>STALL handshake</p> <p>Software can set this bit to make USBFS sends STALL handshake during an OUT transaction. This bit has a higher priority than NAKS bit in this register and GINA K in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect.</p> <p>For control OUT endpoint:</p> <p>Only USBFS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it.</p> <p>For interrupt or bulk OUT endpoint:</p> <p>Only software can clear this bit.</p>
20	SNOOP	<p>Snoop mode</p> <p>This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBFS doesn't check the received data packet's CRC value.</p> <p>0:Snoop mode disabled 1:Snoop mode enabled</p>
19:18	EPTYPE[1:0]	<p>Endpoint type</p> <p>This field defines the transfer type of this endpoint:</p> <p>00: Control 01: Isochronous 10: Bulk 11: Interrupt</p>
17	NAKS	<p>NAK status</p> <p>This bit controls the NAK status of USBFS when both STALL bit in this register and GONS bit in USBFS_DCTL register are cleared:</p> <p>0: USBFS sends handshake packets according to the status of the endpoint's Rx FIFO. 1: USBFS always sends NAK handshake to the OUT token.</p> <p>This bit is read-only and software should use CNAK and SNAK in this register to control this bit.</p>
16	EOFRM	<p>Even/odd frame (For isochronous OUT endpoints)</p> <p>For isochronous transfers, software can use this bit to control that USBFS only receives data packets in even or odd frames. If the current frame number's parity doesn't match with this bit, USBFS just drops the data packet.</p> <p>0: Only sends data in even frames 1: Only sends data in odd frames</p>
	DPID	Endpoint data PID (For interrupt/bulk OUT endpoints)

These is a data PID toggle scheme in interrupt or bulk transfer. Softw are should set SD0PID to set this bit before a transfer starts and USBFS maintains this bit during transfers following the data toggle scheme described in USB protocol.

0: Data packet's PID is DATA0

1: Data packet's PID is DATA1

15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

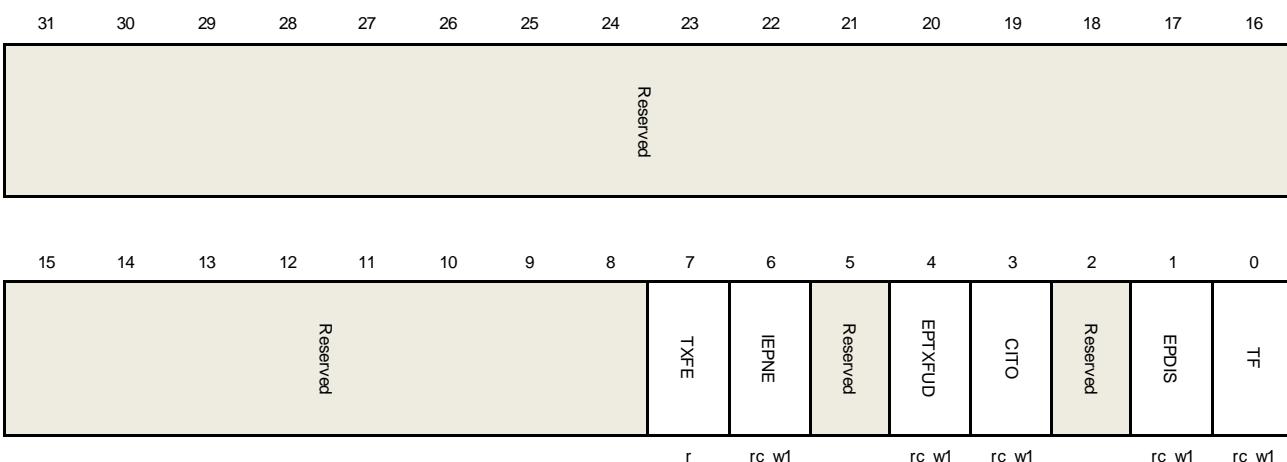
Device IN endpoint-x interrupt flag register (USBFS_DIEPxINTF) (x=0..3, where x = endpoint_number)

Address offset: 0x0908 + (endpoint_number × 0x20)

Reset value: 0x0000 0080

This register contains the status and events of an IN endpoint, when an IN endpoint interrupt occurs, read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1 except the read-only TXFE bit.

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	TXFE	Transmit FIFO empty The Tx FIFO of this IN endpoint has reached the empty threshold value defined by TXFTH field in USBFS_GAHBCS register.
6	IEPNE	IN endpoint NAK effective

The setting of SNAK bit in USBFS_DIEPxCTL register takes effect. This bit can be cleared either by writing 1 to it or by setting CNAK bit in USBFS_DIEPxCTL register.

5	Reserved	Must be kept at reset value
4	EPTXFUD	Endpoint Tx FIFO underrun This flag is triggered if the Tx FIFO has no packet data when an IN token is incoming
3	CITO	Control In Timeout interrupt This flag is triggered if the device waiting for a handshake is timeout in a control IN transaction.
2	Reserved	Must be kept at reset value
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled by the software's request.
0	TF	Transfer finished This flag is triggered when all the IN transactions assigned to this endpoint have been finished.

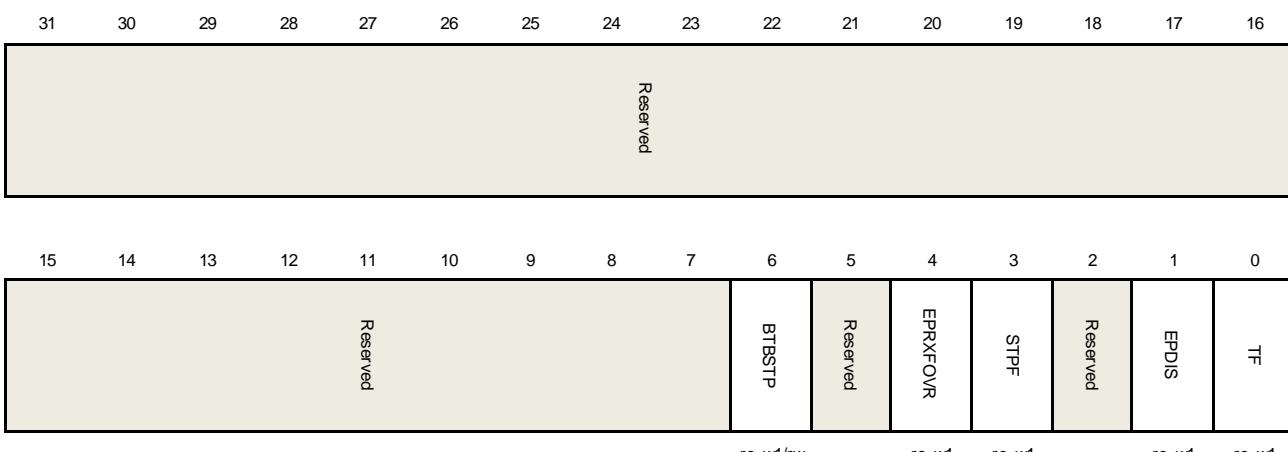
Device OUT endpoint-x interrupt flag register (USBFS_DOEPxINTF) (x = 0..3, where x = endpoint_number)

Address offset: 0x0B08 + (endpoint_number × 0x20)

Reset value: 0x0000 0000

This register contains the status and events of an OUT endpoint, when an OUT endpoint interrupt occurs, read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value

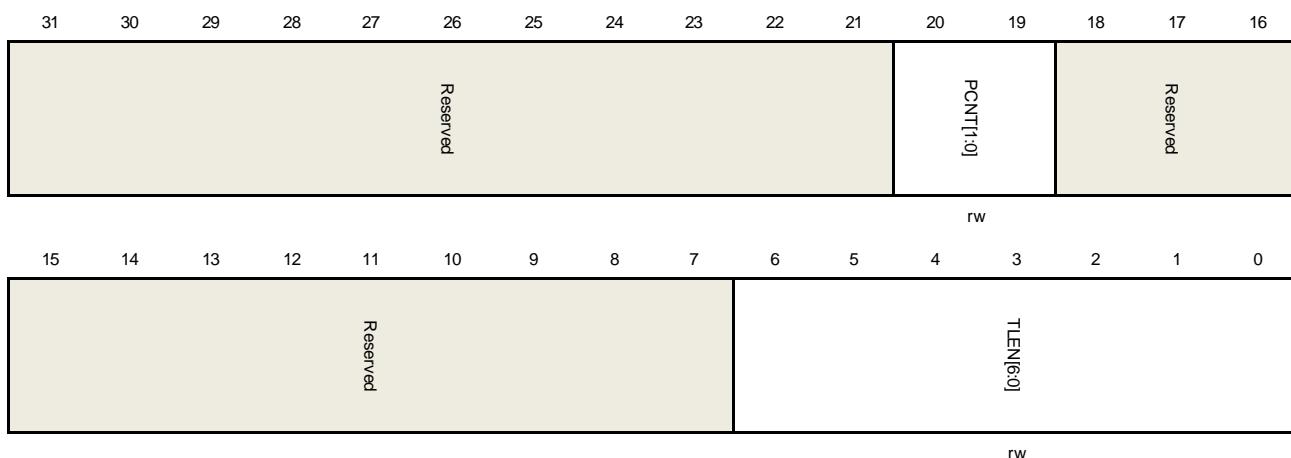
6	BTBSTP	Back-to-back SETUP packets (Only for control OUT endpoint) This flag is triggered when a control out endpoint has received more than 3 back-to-back setup packets.
5	Reserved	Must be kept at reset value
4	EPRXFOVR	Endpoint Rx FIFO overrun This flag is triggered if the OUT endpoint's Rx FIFO has no enough space for a packet data when an OUT token is incoming. USBFS will drop the incoming OUT data packet and sends a NAK handshake in this case.
3	STPF	SETUP phase finished (Only for control OUT endpoint) This flag is triggered when a setup phase finished, i.e. USBFS receives an IN or OUT token after a setup token.
2	Reserved	Must be kept at reset value
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled by the software's request.
0	TF	Transfer finished This flag is triggered when all the OUT transactions assigned to this endpoint have been finished.

Device IN endpoint 0 transfer length register (USBFS_DIEP0LEN)

Address offset: 0x0910

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value
20:19	PCNT[1:0]	Packet count

The number of data packets desired to be transmitted in a transfer.

Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.

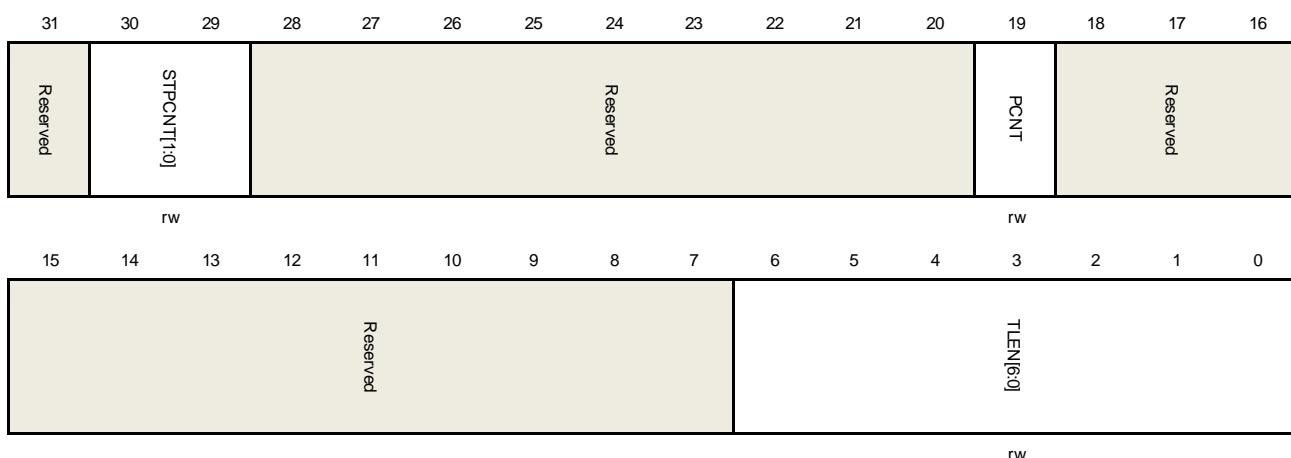
18:7	Reserved	Must be kept at reset value
6:0	TLEN[6:0]	<p>Transfer length</p> <p>The total data bytes number of a transfer.</p> <p>This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Program this field before the endpoint is enabled. When software successfully writes a packet into the endpoint's Tx FIFO, this field is decreased by the byte size of the packet.</p>

Device OUT endpoint 0 transfer length register (USBFS_DOEP0LEN)

Address offset: 0x0B10

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:29	STPCNT[1:0]	<p>SETUP packet count</p> <p>This field defines the maximum number of back-to-back SETUP packets this endpoint can accept.</p> <p>Program this field before setup transfers. Each time a back-to-back setup packet is received, USBFS decrease this field by one. When this field reaches zero, the BTBSTP flag in USBFS_DOEP0INTF register will be triggered.</p> <p>00: 0 packet 01: 1 packet 10: 2 packets</p>

11: 3 packets

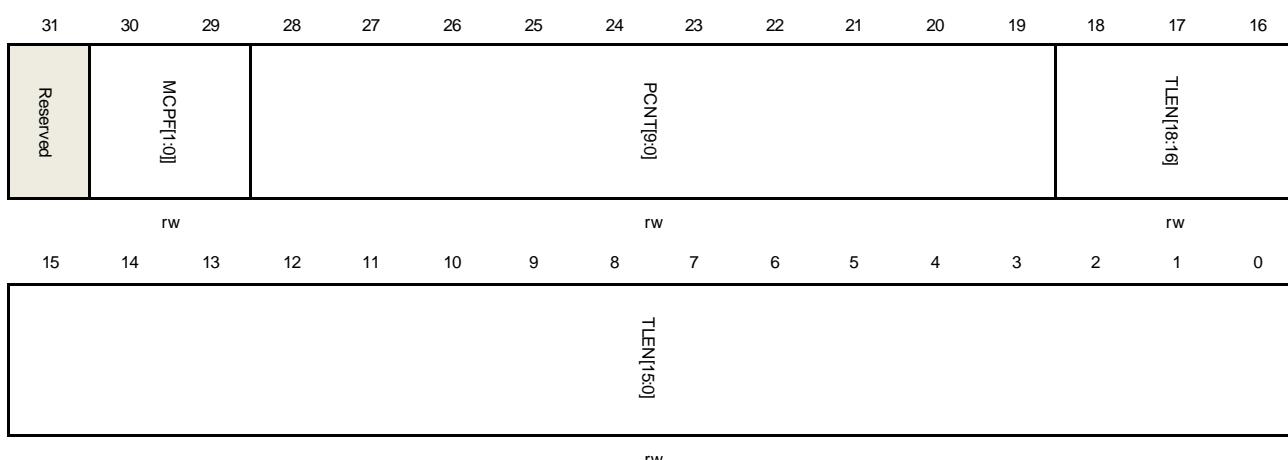
28:20	Reserved	Must be kept at reset value
19	PCNT	<p>Packet count</p> <p>The number of data packets desired to receive in a transfer.</p> <p>Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet reception on bus.</p>
18:7	Reserved	Must be kept at reset value
6:0	TLEN[6:0]	<p>Transfer length</p> <p>The total data bytes number of a transfer.</p> <p>This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Program this field before the endpoint is enabled. Each time software reads out a packet from the Rx FIFO, this field is decreased by the byte size of the packet.</p>

**Device IN endpoint-x transfer length register (USBFS_DIEPxLEN) (x = 1..3,
where x = endpoint_number)**

Address offset: 0x910 + (endpoint_number × 0x20)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:29	MCPF[1:0]	<p>Multi packet count per frame</p> <p>This field indicates the packet count that must be transmitted per frame for periodic IN endpoints on the USB. It is used to calculate the data PID for isochronous IN endpoints by the core.</p> <p>01: 1 packet</p>

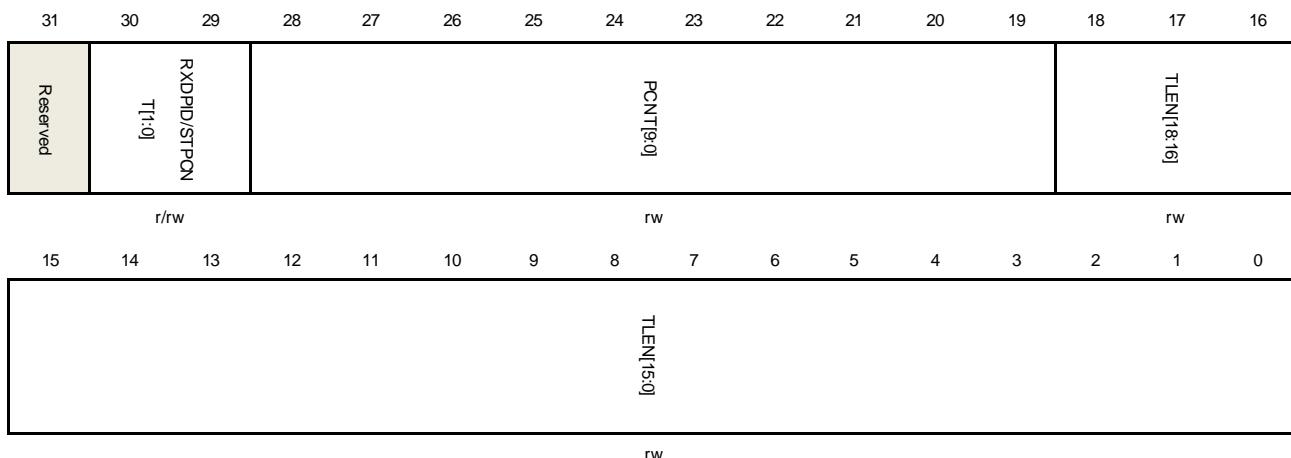
		10: 2 packets 11: 3 packets
28:19	PCNT[9:0]	<p>Packet count</p> <p>The number of data packets desired to be transmitted in a transfer.</p> <p>Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.</p>
18:0	TLEN[18:0]	<p>Transfer length</p> <p>The total data bytes number of a transfer.</p> <p>This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Program this field before the endpoint is enabled. When software successfully writes a packet into the endpoint's Tx FIFO, this field is decreased by the byte size of the packet.</p>

Device OUT endpoint-x transfer length register (USBFS_DOEPxLEN) (x = 1..3, where x = endpoint_number)

Address offset: 0x0B10 + (endpoint_number × 0x20)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:29	RXDPID[1:0]	<p>Received data PID (For isochronous OUT endpoints)</p> <p>This field saves the PID of the latest received data packet on this endpoint.</p> <p>00: DATA0 10: DATA1 Others: Reserved</p>
	STPCNT[1:0]	SETUP packet count (For control OUT Endpoints.)

This field defines the maximum number of back-to-back SETUP packets this endpoint can accept.

Program this field before setup transfers. Each time a back-to-back setup packet is received, USBFS decrease this field by one. When this field reaches zero, the BTBSTP flag in USBFS_DOEPxINTF register will be triggered.

00: 0 packet

01: 1 packet

10: 2 packets

11: 3 packets

28:19	PCNT[9:0]	Packet count The number of data packets desired to receive in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet reception on bus.
18:0	TLEN[18:0]	Transfer length The total data bytes number of a transfer. This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Program this field before the endpoint is enabled. Each time after software reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.

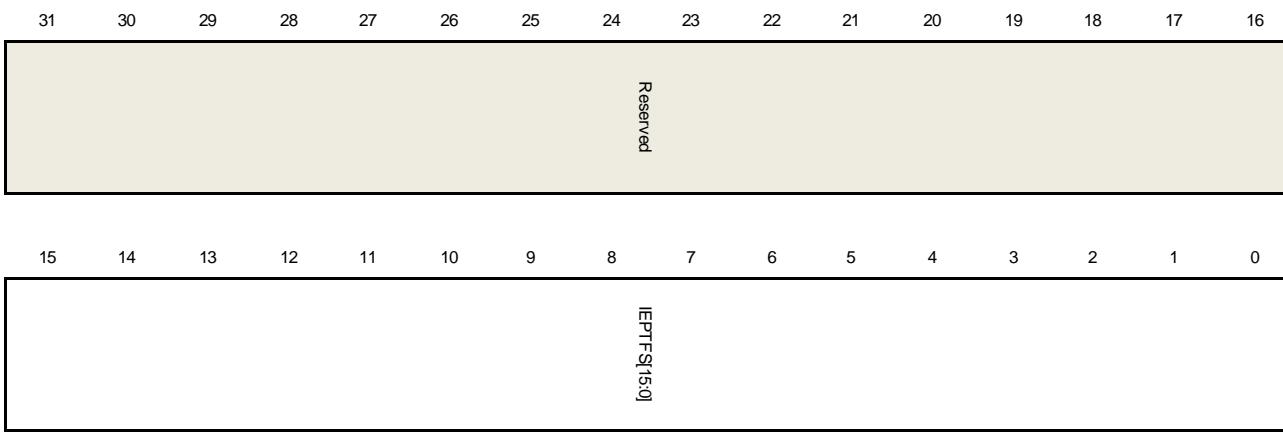
Device IN endpoint-x transmit FIFO status register (USBFS_DIEPxTFSTAT) (x = 0..3, where x = endpoint_number)

Address offset: 0x0918 + (endpoint_number × 0x20)

Reset value: 0x0000 0200

This register contains the information of each endpoint's Tx FIFO.

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



r

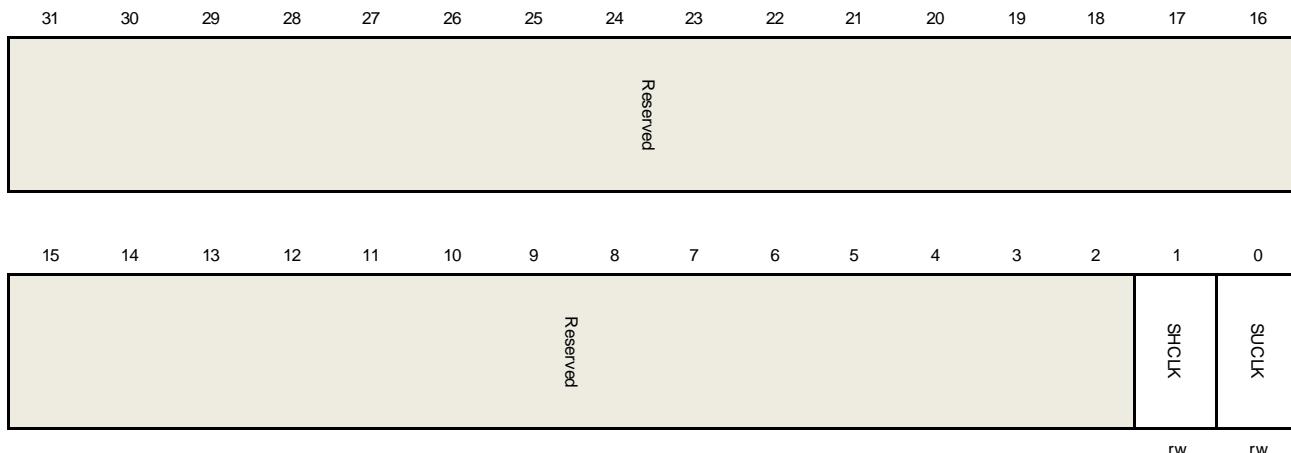
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	IEPTFS[15:0]	IN endpoint's Tx FIFO space remaining IN endpoint's Tx FIFO space remaining in 32-bit words: 0: FIFO is full 1: 1 word available ... n: n words available

24.7.4. Power and clock control register (USBFS_PWRCLKCTL)

Address offset: 0x0E00

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	SHCLK	Stop HCLK Stop the HCLK to save power. 0:HCLK is not stopped 1:HCLK is stopped
0	SUCLK	Stop the USB clock Stop the USB clock to save power. 0:USB clock is not stopped 1:USB clock is stopped

25. Digital camera interface (DCI)

25.1. Overview

DCI is a parallel interface to capture video or picture from a camera. It supports various color space such as YUV/RGB, as well as compression format such as JPEG.

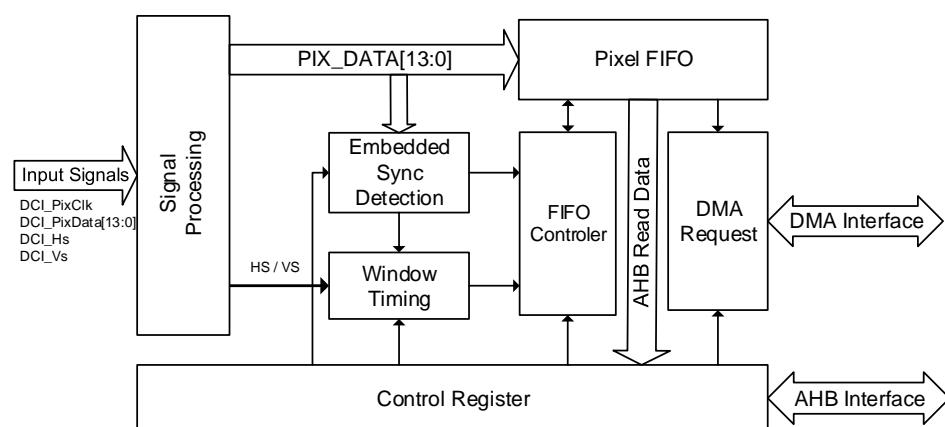
25.2. Characteristics

- Digital video/picture capture
- 8/10/12/14 data width supported
- High transfer efficiency with DMA interface
- Video/picture crop supported
- Various pixel digital encoding formats supported including YCbCr422/RGB565
- JPEG compression format supported
- Hard/embedded synchronous signals supported

25.3. Block diagram

The DCI contains these modules: Signal Processing, Pixel FIFO, FIFO controller, window timing, embedded sync detection, DMA interface and control register.

Figure 25-1. DCI module block diagram



The signal processing module generates useful signals for other internal modules from external input signals. The frequency of HCLK should be 2.5 times higher than DCI_PixClk to ensure the proper operation of signal processing module.

The embedded sync detection module is designed to support embedded synchronization mode. In DCI embedded synchronization mode, video synchronization information is embedded into pixel data and there is no hardware horizontal or vertical synchronization

signal (DCI_Hs or DCI_Vs). DCI uses embedded sync detection module to extract synchronization information from pixel data, and then recover horizontal and vertical synchronization signals.

The window timing module performs image cutting function. This module calculates a pixel's position using synchronization signals either from DCI interface or embedded sync detection module and then decides whether this pixel data needs to be received according to the configuration of DCI_CWSPOS and DCI_CWSZ registers.

DCI uses a 4 word (32-bit) FIFO to buffer the received pixel data. If DMA mode is enabled, the DMA interface asserts a DMA request every time the FIFO is not empty. Control register provides register interface between DCI and software.

25.4. Signal description

Table 25-1. PINs used by DCI

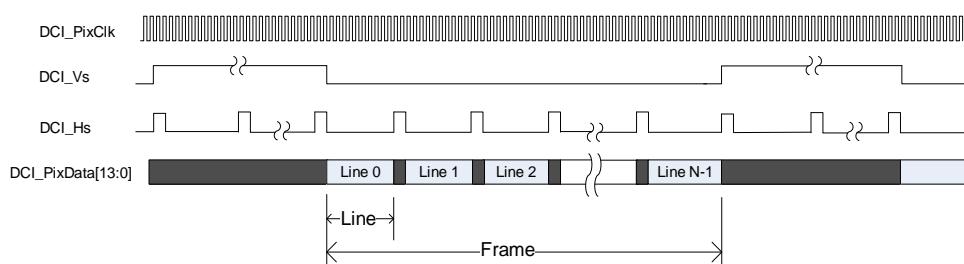
Direction	Name	Width	Description
Input	DCI_PixClk	1	DCI Pixel Clock
Input	DCI_PixData	14	DCI Pixel Data
Input	DCI_Hs	1	DCI Horizontal Synchronization
Input	DCI_Vs	1	DCI Vertical Synchronization

25.5. Function overview

25.5.1. DCI hardware synchronization mode

In DCI hardware synchronization mode (ESM bit in DCI_CTL register is 0), DCI_Hs and DCI_Vs signals are used to indicate the start of a line and a frame. DCI captures pixel data from DCI_PixData[13:0] at rising or falling edge of DCI_PixClk (clock polarity is configured by CKS bit in DCI_CTL).

Figure 25-2. Hardware synchronization mode

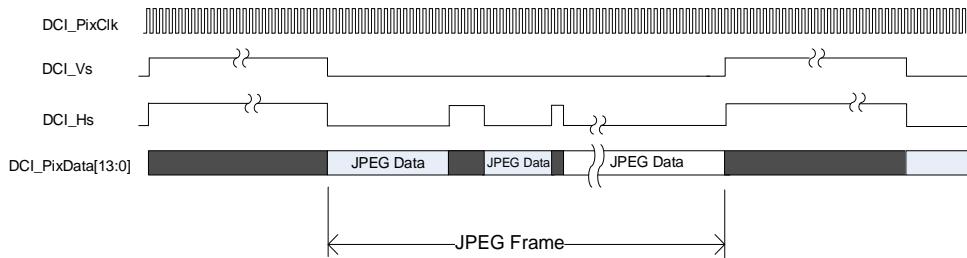


The above figure assumes that the polarities of both DCI_Hs and DCI_Vs are high during blanking period, so the data on DCI_PixData lines is only valid when both DCI_Hs and DCI_Vs are low.

JPEG mode

DCI supports JPEG video/picture compression format in hardware synchronization mode. In JPEG mode (JM bit in DCI_CTL is set), the DCI_Vs is used to indicate start of a new frame, and DCI_Hs is used as stream data valid signal.

Figure 25-3. Hardware synchronization mode: JPEG format supporting



25.5.2. Embedded synchronization mode

DCI supports embedded synchronization mode. In this mode there are only DCI_PixData and DCI_PixClk signals in DCI interface and the synchronization information is embedded in the pixel data. This mode is enabled by setting ESM bit and clearing JM bit in DCI_CTL register.

In embedded synchronization mode, line and frame synchronization information is encoded into sync code and embedded into the pixel data. There are four kinds of sync code: Line Start(LS), Line End(LE), Frame Start(FS) and Frame End(FE). In this mode the data width is forced to 8 and each sync code is composed by 4-byte sequence: FF-00-00-MN, and MN is defined in DCI_SC register. In embedded synchronization mode, the 0xFF and 0x00 should not appear in pixel data to avoid mistake.

In embedded synchronization mode, DCI starts to detect the sync codes after enabled and recover line/frame synchronization information. For example, DCI starts to capture a new frame if it detects a Frame End code and then a Frame Start Code.

When detecting sync code, it is possible to make DCI compare only a few bits of MN byte in FF_00_00_MN sequence by configuring sync code unmask register (DCI_SCUMSK). DCI will only compare bits unmasked by DCI_SCUMSK register. For example: LS in DCI_SC register is A5 and LSM in DCI_SCUMSK is F0, then DCI will only compare the higher 4 bits for LS sync code and thus, FF-00-00-A6 sequence will also be detected as a LS code.

25.5.3. Capture data using snapshot or continuous capture modes

The DCI supports two capture modes: snapshot and continuous capture. Capture mode is configured by SNAP bit in DCI_CTL register.

After correctly configure, enable DCI and set CAP bit in DCI_CTL register, the DCI begins to detect frame start. It begins to capture data once a frame start is detected. In snapshot mode(SNAP=1), DCI automatically stops capturing and clears the CAP bit after a whole frame is captured completely, while in continuous mode, DCI prepares to capture the next frame.

The DCI capture frequency is defined by FR[1:0] bits in continuous mode. For example, if FR[1:0]=00, DCI captures each frame, and if FR[1:0]=01, DCI only captures every alternate frame.

In continuous mode, software may clear the CAP bit any time when DCI is capturing data, but DCI doesn't stop capture immediately. It always stops after a complete frame ends. Software should read back the CAP bit to know whether the DCI stops effectively.

25.5.4. Window function

DCI supports window function which is able to cut a part of image from the captured frame. Window function is enabled by setting WDEN bit in DCI_CTL register and this function is disabled in JPEG mode.

DCI continuously counts and calculates pixels' horizontal and vertical position during capturing, and compares the position and the values in crop window registers (DCI_CWSPOS and DCI_CWSZ), and then discards those pixels outside the crop window and only pushes pixels inside the window into the pixel FIFO.

If a frame ends when the vertical lines size defined in DCI_CWSZ is not reached yet, the end of frame flag will be triggered and DCI stops the capture.

25.5.5. Pixel formats, data padding and DMA

DCI supports various pixel digital encoding formats including YCbCr422/RGB565. However, DCI only receives these pixel data, pads these pixels into a word and push into a pixel FIFO. DCI doesn't perform any pixel format conversion or data processing and doesn't care about the detail of pixel format.

DCI uses a 32-bits width data buffer to transfer between DCI interface and pixel FIFO. There are two padding method in this module: byte padding and half-word padding, depending on the data width of DCI interface. Data width is configured by DCIF[1:0] in DCI_CTL register. The data width is fixed to 8 in JPEG mode and embedded synchronization mode.

The DMA interface sends DMA request when FIFO is not empty.

Byte padding mode

Byte padding mode is used if data width of DCI interface is 8. In byte padding mode four bytes are filled into the 32-bits width data buffer. In Non-JPEG mode, the DCI pushes the 32-bits buffer's data into the pixel FIFO when the buffer is full or meets the end of a line. In JPEG mode, the DCI pushes the 32-bits buffer's data into the pixel FIFO when the buffer is full or meets the end of a frame.

Table 25-2. Memory view in byte padding mode

D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]
---------	---------	---------	---------

D7[7:0]	D6[7:0]	D5[7:0]	D4[7:0]
---------	---------	---------	---------

Half-word padding mode

Half-word padding is used if data width of DCI interface is configured into 10/12/14. In this mode each pixel data is extended into 16-bits length by filling zero at higher position, so the 32-bits width data buffer is able to hold two pixel data. DCI pushes the data buffer into pixel FIFO each time the buffer is full or line end.

Table 25-3. Memory view in half-word padding mode

2'b00	D1[13:0]	2'b00	D0[13:0]
2'b00	D3[13:0]	2'b00	D2[13:0]
2'b00	D5[13:0]	2'b00	D4[13:0]
2'b00	D7[13:0]	2'b00	D6[13:0]

25.6. Interrupts

There are several status and error flags in DCI, and interrupts may be asserted from these flags. These status and error flags will assert global DCI interrupt if enabled by corresponding bit in DCI_INTEN. These flags are cleared by writing into DCI_INTC register.

Table 25-4. Status/Error flags

Status Flag Name	Description
ELF	End of Line Flag
EFF	End of Frame Flag
OVRF	FIFO Overrun Flag
VSF	Frame VS Blank Flag
ESEF	Embedded Sync Error Flag

25.7. Register definition

DCI secure access base address: 0x5C05 0000
 DCI non-secure access base address: 0x4C05 0000

25.7.1. Control register (DCI_CTL)

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																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	DCIEN	Reserved	DCIF[1:0]	FR[1:0]	VPS	HPS	CKS	ESM	JM	WDEN	SNAP	CAP	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	DCIEN	DCI Enable 0: DCI is disabled 1: DCI is enabled
13:12	Reserved	Must be kept at reset value
11:10	DCIF[1:0]	Digital Camera Interface Format 00: 8-bit data on every pixel clock 01: 10-bit data on every pixel clock 10: 12-bit data on every pixel clock 11: 14-bit data on every pixel clock
9:8	FR[1:0]	Frame Rate FR defines the frame capture rate in continuous capture mode 00: Capture All frames 01: Capture one in 2 frames 10: Capture one in 4 frames 11: reserved
7	VPS	Vertical Polarity Selection 0: Low level during blanking period 1: High level during blanking period
6	HPS	Horizontal Polarity Selection 0: Low level during blanking period

		1: High level during blanking period
5	CKS	Clock Polarity Selection 0: Capture at falling edge 1: Capture at rising edge
4	ESM	Embedded Synchronous Mode 0: Embedded synchronous mode is disabled 1: Embedded synchronous mode is enabled
3	JM	JPEG Mode 0: JPEG mode is disabled 1: JPEG mode is enabled
2	WDEN	Window Enable 0: Window is disabled 1: Window is enabled
1	SNAP	Snapshot Mode 0: Continuous capture mode 1: Snapshot capture mode
0	CAP	Capture Enable 0: Frame not captured 1: Frame is captured

25.7.2. Status register0 (DCI_STAT0)

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	FV	FIFO Valid 0: No valid pixel data in FIFO 1: Valid pixel data in FIFO
1	VS	VS line status

		0: Not in vertical blanking period 1: In vertical blanking period
0	HS	HS line status 0: Not in horizontal blanking period 1: In horizontal blanking period

25.7.3. Status register1 (DCI_STAT1)

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
					Reserved					ELF	VSF	ESEF	OVRF	EFF	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	ELF	End of Line Flag 0 : No end of line flag 1: A line is captured by DCI
3	VSF	Vsync Flag 0: No vsync flag 1: A vsync blanking detected
2	ESEF	Embedded Synchronous Error Flag 0: No Embedded Synchronous Error Flag 1: A Embedded Synchronous Error detected
1	OVRF	FIFO Overrun Flag 0: No FIFO Overrun 1: A FIFO overrun occurs
0	EFF	End of Frame Flag 0: No end of frame flag 1: A frame is captured by DCI

25.7.4. Interrupt enable register (DCI_INTEN)

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
Reserved										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	ELIE	End of Line Interrupt Enable 0: End of line flag w on't generate interrupt 1: End of line flag w ill generate interrupt
3	VSIE	Vsync Interrupt Enable 0: Vsync flag w on't generate interrupt 1: Vsync flag w ill generate interrupt
2	ESEIE	Embedded Synchronous Error Interrupt Enable 0: Embedded Synchronous Error Flag w on't generate interrupt 1: Embedded Synchronous Error Flag w ill generate interrupt
1	OVRIE	FIFO Overrun Interrupt Enable 0: FIFO Overrun w on't generate interrupt 1: FIFO Overrun w ill generate interrupt
0	EFIE	End of Frame Interrupt Enable 0: End of frame flag w on't generate interrupt 1: End of frame flag w ill generate interrupt

25.7.5. Interrupt flag register (DCI_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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										r	r	r	r	r	r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	ELIF	End of Line Interrupt Flag
3	VSIF	Vsync Interrupt Flag
2	ESEIF	Embedded Synchronous Error Interrupt Flag
1	OVRIF	FIFO Overrun Interrupt Flag
0	EFIF	End of Frame Interrupt Flag

25.7.6. Interrupt flag clear register (DCI_INTC)

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										ELFC	VSFC	ESEFC	OVRFC	EFFC	
										w	w	w	w	w	w

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	ELFC	End of Line Flag Clear Write 1 to clear end of line flag
3	VSFC	Vsync flag clear Write 1 to clear vsync flag
2	ESEFC	Clear embedded synchronous Error Flag Write 1 to clear Embedded Synchronous Error Flag
1	OVRFC	Clear FIFO Overrun Flag Write 1 to clear FIFO Overrun flag
0	EFFC	Clear End of Frame Flag Write 1 to clear end of frame flag

25.7.7. Synchronization codes register (DCI_SC)

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

Bits	Fields	Descriptions
31:24	FE[7:0]	Frame End Code in Embedded Synchronous Mode
23:16	LE[7:0]	Line End Code in Embedded Synchronous Mode
15:8	LS[7:0]	Line Start Code in Embedded Synchronous Mode
7:0	FS[7:0]	Frame Start Code in Embedded Synchronous Mode

25.7.8. Synchronization codes unmask register (DCI_SCUMSK)

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

Bits	Fields	Descriptions
31:24	FEM[7:0]	Frame End Code unMask Bits in Embedded Synchronous Mode
23:16	LEM[7:0]	Line End Code unMask Bits in Embedded Synchronous Mode
15:8	LSM[7:0]	Line Start Code unMask Bits in Embedded Synchronous Mode
7:0	FSM[7:0]	Frame Start Code unMask Bits in Embedded Synchronous Mode

25.7.9. Cropping window start position register (DCI_CWSPOS)

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

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:16	WVSP[12:0]	Window Vertical Start Position Zero means the first line
15:14	Reserved	Must be kept at reset value
13:0	WHSP[13:0]	Window Horizontal Start Position Zero means the first pixel in a line

25.7.10. Cropping window size register (DCI_CWSZ)

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	WVSZ[13:0]	Window Vertical Size WVSZ=X means X+1 lines
15:14	Reserved	Must be kept at reset value.
13:0	WHSZ[13:0]	Window Horizontal Size WHSZ=X means X+1 pixels in a line

25.7.11. DATA register (DCI_DATA)

Address offset: 0x28

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	DT3[7:0]	Pixel Data 3
23:16	DT2[7:0]	Pixel Data 2
15:8	DT1[7:0]	Pixel Data 1
7:0	DT0[7:0]	Pixel Data 0

26. Touch sensing interface (TSI)

26.1. Overview

Touch Sensing Interface (TSI) provides a convenient solution for touch keys, sliders and capacitive proximity sensing applications. The controller builds on charge transfer method. Placing a finger near fringing electric fields adds capacitance to the system and TSI is able to measure this capacitance change using charge transfer method.

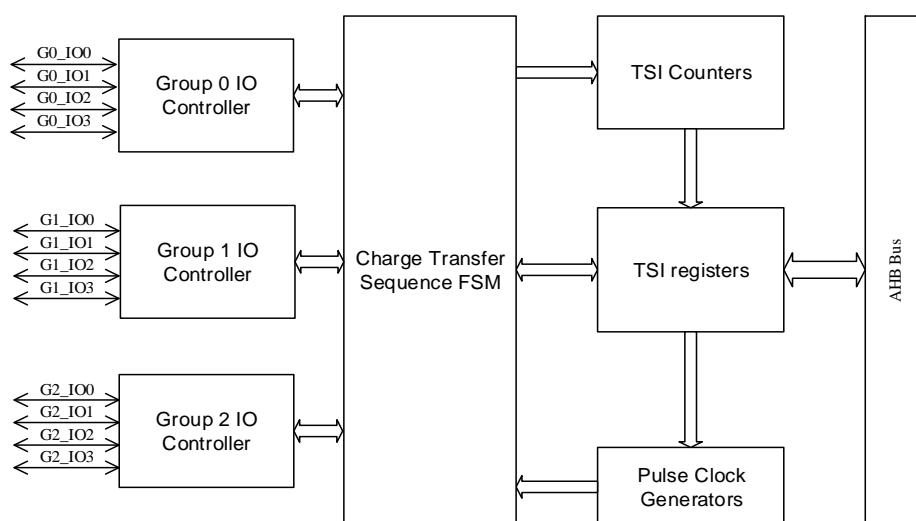
26.2. Characteristics

- Charge transfer sequence fully controlled by hardware.
- 3 fully parallel groups implemented.
- 9 IOs configurable for capacitive sensing Channel Pins and 3 for Sample Pins.
- Configurable transfer sequence frequency.
- Able to implement the user specific charge transfer sequences.
- Sequence end and error flags / configurable interrupts.
- Spread spectrum function implemented.

26.3. Function Overview

26.3.1. TSI block diagram

Figure 26-1. Block diagram of TSI module



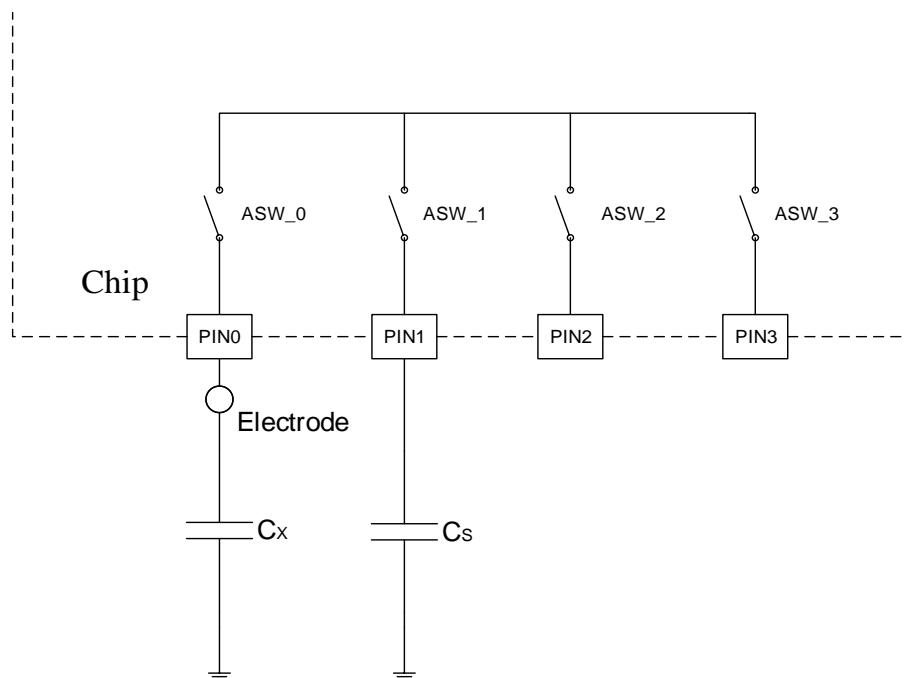
26.3.2. Touch sensing technique overview

There are different technologies for touch sensing, such as optical, resistive, capacitive, strain, 908

etc. Detecting the change of a system is the key problem and goal in these technologies. The TSI module is designed to use charge transfer method which detects the capacitive change of an electrode when touched by or a finger close to it. In order to detect the capacitive change, TSI performs a charge transfer sequence including several charging, transfer steps. The number of these steps indicates the capacitance of an electrode. So the application is able to detect the change of capacitance by monitoring the step number of each transfer sequence.

As shown in [**Figure 26-2. Block diagram of Sample pin and Channel Pin**](#), there are 4 PINs in one group and each PIN has an analog switch connected to a common point which is the key component to implement charge transfer. There should be a sample pin and one or more channel pin(s) configured in one group. In [**Figure 26-2. Block diagram of Sample pin and Channel Pin**](#), PIN0 is a channel pin and PIN1 is a sample pin while PIN2 and PIN3 are unused. An electrode connecting PIN0 is designed on PCB board. The A sample capacitor C_s connected to sample pin PIN1 is also required. Now the capacitance of the channel pin PIN0 includes C_x and the capacitance introduced by the electrode, so capacitance of PIN0 increases when a finger is touching while the capacitance of PIN1 remains unchanged. Thus, the finger's touching can be detected if the capacitance of PIN0 can be measured. In TSI module, a charge-transfer sequence is performed to measure the capacitance of the channel pin(s) in a group, which will be detailed in next section.

Figure 26-2. Block diagram of Sample pin and Channel Pin



26.3.3. Charge transfer sequence

In order to measure the capacitance of a channel pin, charge transfer sequence is performed in chip. The sequence shown in [**Table 26-1. Pin and analog switch state in a charge-transfer sequence**](#) is described based on the connection of [**Figure 26-2. Block diagram of Sample pin and Channel Pin**](#), i.e. PIN0 is channel pin and PIN1 is sample pin.

Table 26-1. Pin and analog switch state in a charge-transfer sequence

Step	Name	ASW_0	ASW_1	Pin0	Pin1
1	Discharge	Close	Close	Input Floating	Pull Down
2	Buffer Time1	Open	Open	Input Floating	Input Floating
3	Charge	Open	Open	Output High	Input Floating
4	Extend Charge	Open	Open	Output High	Input Floating
5	Buffer Time2	Open	Open	Input Floating	Input Floating
6	Charge Transfer	Close	Close	Input Floating	Input Floating
7	Buffer Time3	Open	Open	Input Floating	Input Floating
8	Compare	Open	Open	Input Floating	Input Floating

1. Discharge

Both C_x and C_s are discharged by closing ASW_0 and ASW_1 and configuring PIN1 to pull down. This step is the initial operation for a correct charge transfer sequence and is performed by software before starting a charge transfer sequence. Discharging time in this step should be guaranteed to ensure that the voltage of C_x and C_s are discharged to zero.

2. Buffer Time1

Buffer time with ASW_0 and ASW_1 open, PIN0 is configured to input floating.

3. Charge

Channel pin PIN0 is configured to output high, in order to charge C_x . ASW_0 and ASW_1 remain open during this step. The charging time should be configured (see Register Section for detail) to ensure that the voltage of C_x is charged to V_{DD} .

4. Extend Charge

This is an optional step in a charge-transfer sequence and the behavior of all pins and analog switches in this step is the same as Step 3. The only difference between this and step 3 is the duration time, which is configurable in TSI registers. The duration of this step changes in each loop of a charge-transfer sequence, spreading the spectrum.

5. Buffer Time2

Buffer time with ASW_0 and ASW_1 open, PIN0 is configured to input floating.

6. Charge transfer

ASW_0 and ASW_1 are closed and PIN0 is configured to input floating to transfer charge from C_x to C_s . The transfer time should be configured (see Register Section for detail) to ensure the full transfer after that the voltage of C_x and C_s will be equal.

7. Buffer Time3

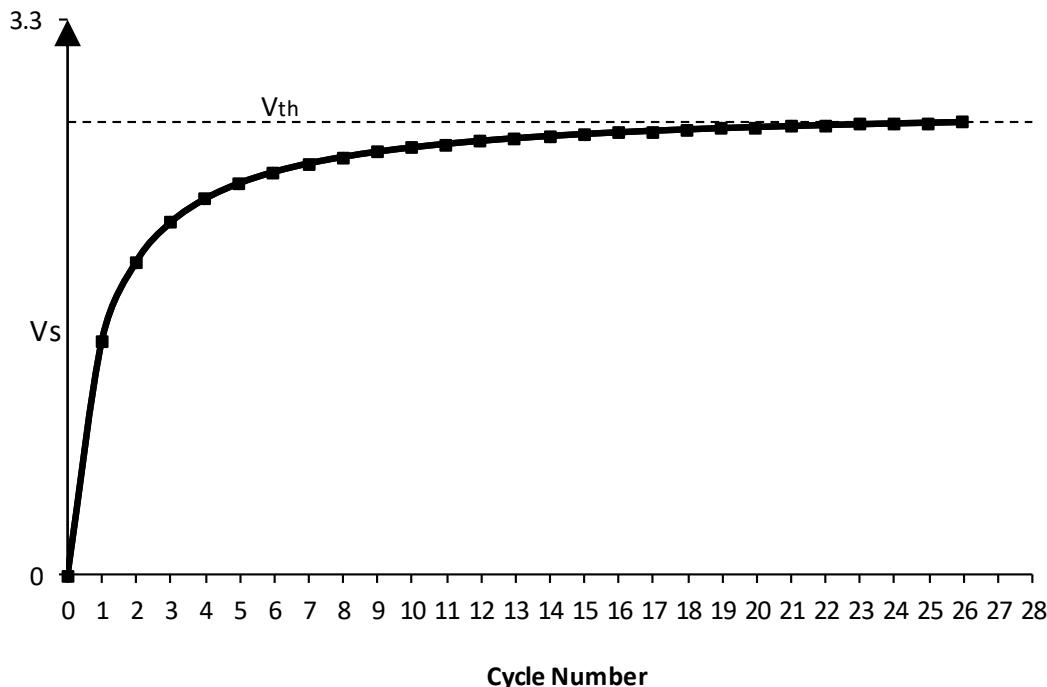
Buffer time with ASW_0 and ASW_1 open, PIN0 is configured to input floating.

8. Compare

ASW_0, ASW_1 and PIN0 remain the configuration of Step7. At this step, the voltage of sample pin PIN1 is compared to a threshold called V_{th} . If voltage of PIN1 is lower than V_{th} , the sequence returns to Step2 and continues, otherwise, the sequence ends.

The voltage of sample pin V_s is zero after initial step and increases after each charge cycle, as shown in [Figure 26-3. Voltage of a sample pin during charge-transfer sequence](#). A larger C_x will cause a greater increase during a cycle. The sequence stops when V_s reaches V_{th} . Each group has a counter which records the number of cycles performed on it to reach V_{th} . At the end of charge-transfer sequence, the group counter is read out to estimate the C_x , i.e. a smaller counter values indicates a larger C_x .

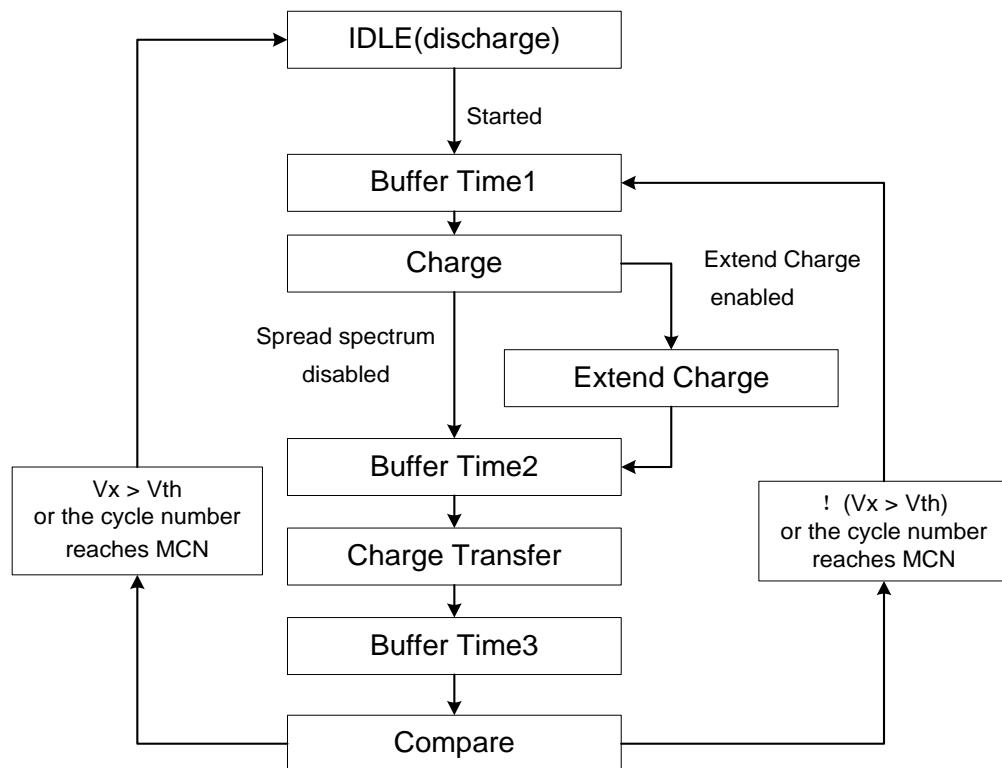
Figure 26-3. Voltage of a sample pin during charge-transfer sequence



26.3.4. Charge transfer sequence FSM

A hardware Finite-state machine (FSM) is designed in chip to perform the charge transfer sequence described in the previous section as shown in [Figure 26-4. FSM flow of a charge-transfer sequence](#).

Figure 26-4. FSM flow of a charge-transfer sequence



This FSM remains in IDLE state after reset. There are 2 kinds of start condition defined by TRGMOD bit in TSI_CTL0 register:

TRGMOD = 0: Software Trigger Mode. In this mode, the FSM starts after TSIS bit in TSI_CTL0 register is written 1 by software.

TRGMOD = 1: Hardware Trigger Mode. In this mode the FSM starts when a falling or rising edge on TSITG pin is detected.

Once started, the FSM runs following the flow described in [Figure 26-4. FSM flow of a charge-transfer sequence](#). The FSM leaves a state if the duration time of this state reaches defined value, and goes into the next state.

The Extend Charge state is present only if the ECEN bit is set in TSI_CTL0 register. This state is designed to implement spread spectrum function, which will extend the duration of the pulse high state with different extend time according to current FSM cycle number. So in other word, the charge frequency becomes dynamic and not fixed. In case of noisy application environment, enabling this function can improve the robustness of TSI. At the same time, system's electromagnetic emissions will be reduced.

In comparing state, the FSM compares voltage of the sample pin in every enabled group and the threshold voltage. If all sample pins' voltage reach the threshold, FSM returns IDLE state and stops, otherwise, it returns to Buffer Time 1 state and begins the next cycle. As shown in [Figure 26-3. Voltage of a sample pin during charge-transfer sequence](#), after 27

cycles, V_s (the voltage of sample pin) reaches V_{th} (the threshold voltage).

There is also a max cycle number defined by MCN in TSI_CTL0 register. When the cycle number reaches MCN, FSM returns to IDLE state and stops after Compare State, whether V_s reaches V_{th} or not.

26.3.5. Clock and duration time of states

There are 3 clocks in TSI module: HCLK, CTCLK(Charge Transfer Clock) and ECCLK(Extend Charge Clock). HCLK is system clock and drives TSI's register and FSM. CTCLK, which is divided from HCLK with division factor defined by register CTCDIV is the clock used for calculating the duration time of the charge state and Charge Transfer state. ECCLK, which is divided from HCLK with division factor defined by register ECCDIV is the clock used to calculate the maximum duration time of Extend Charge state. ECCLK and CTCLK are independent of each other.

The duration time of each state except Extend Charge state is fixed in each loop according to the configuration of the register.

The duration time of Buffer Time1, Buffer Time2 and Buffer Time3 are fixed to 2 HCLK periods. The duration time of Charge state and Charge Transfer state is defined by CDT and CTDT bits (see TSI_CTL0 register section for detail).

Generally, the variation range of extend charge frequency is limited to between 10% and 50%. And the duration time of Extend Charge state changes in each cycle of the charge-transfer FSM and the maximum duration time are defined by ECDT[6:0] in TSI_CTL0 register. If the Extend Charge state is enabled, the longest change time is when cycle number is ECDT+2. The duration time of Extend Charge state in each cycle is presented in [Table 26-2. Duration time of Extend Charge state in each cycle](#).

Table 26-2. Duration time of Extend Charge state in each cycle

Cycle Number	Number of ECCLKs in Extend Charge state
1	0
2	1
...	
ECDT	ECDT-1
ECDT+1	ECDT
ECDT+2	ECDT+1
ECDT+3	ECDT
ECDT+4	ECDT-1
...	...
2*ECDT+1	2
2*ECDT+2	1
2*ECDT+3	0
2*ECDT+4	1
2*ECDT+5	2

Cycle Number	Number of ECCLKs in Extend Charge state
...	...

Table 26-3. Spread spectrum deviation base on HCLK period

HCLK Period	Spread spectrum deviation with different ECDIV value (ECDT=0x7F)		
	ECDIV[2:0]=0x0 (Min)	ECDIV[2:0]=0x1	ECDIV[2:0]=0x7(Max)
41.6ns (24MHz)	5333.3ns	10666.6ns	42666.6ns
20.8ns (48MHz)	2666.6ns	5333.3ns	21333.3ns
13.8ns (72MHz)	1777.7ns	3555.5ns	14222.2ns
11.9ns (84MHz)	1523.8ns	3047.6ns	12190.4ns
9.26ns(108MHz)	1185.18ns	2370.37ns	9481.48ns

26.3.6. PIN mode control of TSI

There are 4 pins in each group and each of these pins is able to be used as a sample pin or channel pin. Only one pin in a group should be configured as sample pin, and channel pins can be more than one. The sample pin and channel pin(s) should not be configured as the same pin in any case.

When a PIN is configured in GPIO (see chapter GPIO) used by TSI, the pin's mode is controlled by TSI. Generally, each pin has 3 modes: input, output high and output low.

The mode of a channel pin or a sample pin during a charge-transfer sequence is described in [Table 26-1. Pin and analog switch state in a charge-transfer sequence](#) which PIN0 represents a channel pin and PIN1 represents a sample pin, i.e. the charge-transfer FSM take control of these channels or sample pins' mode and the states of related analog switches when the sequence is on-going. When the sequence is in IDLE state, PINMOD bit in TSI_CTL0 register defines the mode of these pins. Pins that are configured in GPIO used by TSI but neither sample nor channel in TSI register is called free pins whose mode is defined by PINMOD bit in TSI_CTL0, too.

26.3.7. Analog switch (ASW) and I/O hysteresis mode

A channel or sample pin's analog switch is controlled by charge-transfer sequence when FSM is running, as shown in [Table 26-1. Pin and analog switch state in a charge-transfer sequence](#). When the FSM is IDLE, these pins' analog switches are controlled by GxPy bits in TSI_ASW register. All free pin's analog switches are controlled by GxPy bits too.

TSI takes control of the analog switches when FSM is IDLE, even if these pins are not configured to be used by TSI in GPIO. The user is able to perform user-defined charge-transfer sequence by writing GxPy bits to control these analog switches, while controlling pin mode directly in GPIO.

TSI controller has the highest priority of GPIO. When TSI is enable, this configuration is available regardless of the GPIO mode, controlled by GPIO registers or other peripherals. Disable the GPIO's Schmitt trigger hysteresis of TSI Pins by resetting GxPy bit in TSI_PHM register could improve the system immunity.

26.3.8. TSI operation flow

The normal operation flow of TSI is listed below:

System initialization, such as system clock configuration, TSI related GPIO configuration, etc.

Program TSI_CTL0, TSI_CHCFG, TSI_INTEN, TSI_SAMPCFG and GEx bits of TSI_GCTL register according to demand.

Enable TSI by setting TSIEN bit in TSI_CTL0 register.

Optional for software trigger mode: program TSIS bit to start charging transfer sequence. In hardware trigger mode, TSI is started by falling/rising edge on the trigger pin.

Wait for the CTCF or MNERR flag in TSI_INTF and clear these flags by writing TSI_INTC.

Read out the CYCN bits in TSI_GxCYCN registers.

26.3.9. TSI flags and interrupts

Table 26-4. TSI errors and flags

Flag Name	Description	Cleared by
CTCF	TSI stops because all enabled samplers' sample pins reach V_{th} .	CCTCF bit in TSI_INTC
MNERR	TSI stops because the cycle number reaches the maximum value.	CMNERR bit in TSI_INTC

26.3.10. TSI GPIOs

Table 26-5. TSI pins

TSI 组	TSI 引脚	GPIO 引脚
第 0 引脚组	PIN0	PA0
	PIN1	PA1
	PIN2	PA2
	PIN3	PA3
第 1 引脚组	PIN0	PB0
	PIN1	PB1
	PIN2	PB2
	PIN3	PB10
第 2 引脚组	PIN0	PB11
	PIN1	PB12
	PIN2	PB13
	PIN3	PB14

26.4. Registers definition

TSI secure access base address: 0x5002 4000

TSI non-secure access base address: 0x4002 4000

26.4.1. Control register0 (TSI_CTL0)

Address offset: 0x000

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
CDT[3:0]				CTDT[3:0]				ECDT[6:0]				ECEN			
rw				rw				rw			rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECDIV[0]	CTCDIV[2:0]		Reserved				MCN[2:0]		PINMOD	EGSEL	TRGMOD	TSIS	TSIEN		
rw	rw						rw		rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:28	CDT[3:0]	Charge State Duration Time CDT[3:0] is set and clear by software. These bits controls the duration time of Charge State in a charge-transfer sequence. 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ 1111: $16 \times t_{CTCLK}$
27:24	CTDT[3:0]	Charge Transfer State Duration Time CTDT[3:0] is set and clear by software. These bits control the duration time of Charge Transfer State in a charge-transfer sequence. 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ 1111: $16 \times t_{CTCLK}$
23:17	ECDT[6:0]	Extend Charge State Maximum Duration Time ECDT[6:0] is set and clear by software. These bits control the maximum duration time of Extend Charge Transfer State in a charge-transfer sequence. Extend Charge State is only present when ECEN bit in TSI_CTL0 register is set. 0000000: $1 \times t_{ECCLK}$ 0000001: $2 \times t_{ECCLK}$ 0000010: $3 \times t_{ECCLK}$

	
		1111111: $128 \times t_{ECCLK}$
16	ECEN	<p>Extend Charge State Enable.</p> <p>0: Extend Charge disabled</p> <p>1: Extend Charge enabled</p>
15	ECDIV[0]	<p>Extend Charge clock(ECCLK) division factor.</p> <p>ECCLK in TSI is divided from HCLK and ECDIV defines the division factor.</p> <p>0x0: $f_{ECCLK}=f_{HCLK}$</p> <p>0x1: $f_{ECCLK}=f_{HCLK}/2$</p> <p>0x2: $f_{ECCLK}=f_{HCLK}/3$</p> <p>0x3: $f_{ECCLK}=f_{HCLK}/4$</p> <p>0x4: $f_{ECCLK}=f_{HCLK}/5$</p> <p>0x5: $f_{ECCLK}=f_{HCLK}/6$</p> <p>0x6: $f_{ECCLK}=f_{HCLK}/7$</p> <p>0x7: $f_{ECCLK}=f_{HCLK}/8$</p> <p>Note: ECDIV[2:1] are located in TSI_CTL1 and ECDIV[0] is located in TSI_CTL0.</p>
14:12	CTCDIV[2:0]	<p>Charge Transfer clock(CTCLK) division factor.</p> <p>CTCLK in TSI is divided from HCLK and CTCDIV defines the division factor.</p> <p>0000: $f_{CTCLK}=f_{HCLK}$</p> <p>0001: $f_{CTCLK}=f_{HCLK}/2$</p> <p>0010: $f_{CTCLK}=f_{HCLK}/4$</p> <p>0011: $f_{CTCLK}=f_{HCLK}/8$</p> <p>....</p> <p>0111: $f_{CTCLK}=f_{HCLK}/128$</p> <p>1000: $f_{CTCLK}=f_{HCLK}/256$</p> <p>1001: $f_{CTCLK}=f_{HCLK}/512$</p> <p>....</p> <p>1110: $f_{CTCLK}=f_{HCLK}/16384$</p> <p>1111: $f_{CTCLK}=f_{HCLK}/32768$</p> <p>Note: CTCDIV[3] is located in TSI_CTL1 and CTCDIV[2:0] are located in TSI_CTL0.</p>
11:8	Reserved	Must be kept at reset value
7:5	MCN[2:0]	<p>Max cycle number of a sequence</p> <p>MCN[2:0] defines the max cycle number of a charge-transfer sequence FSM which stops after reaching this number.</p> <p>000: 255</p> <p>001: 511</p> <p>010: 1023</p> <p>011: 2047</p> <p>100: 4095</p> <p>101: 8191</p> <p>110: 16383</p>

		111: Reserved
4	PINMOD	<p>Pin mode</p> <p>This bit defines a TSI pin's mode when charge-transfer sequence is IDLE.</p> <p>0: TSI pin will output low when IDLE</p> <p>1: TSI pin will keep input mode when IDLE</p>
3	EGSEL	<p>Edge selection</p> <p>This bit defines the edge type in hardware trigger mode.</p> <p>0: Falling edge</p> <p>1: Rising edge</p>
2	TRGMOD	<p>Trigger mode selection</p> <p>0: Software Trigger Mode, sequence will start after TSIS bit is set.</p> <p>1: Hardware Trigger Mode, sequence will start after a falling/rising edge on trigger pin detected.</p>
1	TSIS	<p>TSI start</p> <p>This bit is set by software to start a charge-transfer sequence in software trigger mode and reset by hardware when the sequence stops. After setting this bit, software can reset it to stop the started sequence manually.</p> <p>0: TSI is not started</p> <p>1: TSI is started.</p>
0	TSIEN	<p>TSI enable</p> <p>0: TSI module is enabled</p> <p>1: TSI module is disabled</p>

26.4.2. Interrupt enable register(TSI_INTEN)

Address offset: 0x04

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												MNERRIE	CTCFIE		

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	MNERRIE	Max Cycle Number Error Interrupt Enable

0: MNERR interrupt is disabled

1: MNERR interrupt is enabled

0	CTCFIE	Charge-transfer complete flag Interrupt Enable
		0: CTCF interrupt is disabled
		1: CTCF interrupt is enabled

26.4.3. Interrupt flag clear register (TSI_INTC)

Address offset: 0x08

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															CMNERR CCTCF
															w w

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	CMNERR	Clear max cycle number error
		0: Reserved
		1: Clear MNERR
0	CCTCF	Clear charge-transfer complete flag
		0: Reserved
		1: Clear CTCF

26.4.4. Interrupt flag register (TSI_INTF)

Address offset: 0x0C

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															MNERR CTCF
															r r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	MNERR	<p>Max Cycle Number Error</p> <p>This bit is set by hardware after charge-transfer sequence stops because it reaches the max cycle number defined by MCN[2:0]. This bit is cleared by writing 1 to CMNERR bit in TSI_ICR register.</p> <p>0: No Max Count Error</p> <p>1: Max Count Error</p>
0	CTCF	<p>Charge-Transfer complete flag</p> <p>This bit is set by hardware after charge-transfer sequence stops because all enabled group's sample pins reach the threshold voltage or because the cycle number reaches the value defined by MCN[2:0]. This bit is cleared by writing 1 to CCTCF bit in TSI_ICR register.</p> <p>0: Charge-Transfer not complete</p> <p>1: Charge-Transfer complete</p>

26.4.5. Pin hysteresis mode register(TSI_PHM)

Address offset: 0x10

Reset value: 0xFFFF FFFF

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		G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0		
		rw	rw	rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	GxPy	<p>Pin hysteresis mode</p> <p>This bit is set and cleared by software.</p> <p>0: Pin GxPy Schmitt trigger hysteresis mode disabled</p> <p>1: Pin GxPy Schmitt trigger hysteresis mode enabled</p>

26.4.6. Analog switch register(TSI_ASW)

Address offset: 0x18

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				G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0

rw rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	GxPy	Analog switch state. This bit is set and cleared by software. 0: Analog switch of GxPy is open 1: Analog switch of GxPy is closed

26.4.7. Sample configuration register(TSI_SAMPCFG)

Address offset: 0x20

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				G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0

rw rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	GxPy	Sample pin mode This bit is set and cleared by software. 0: Pin GxPy is not a sample pin 1: Pin GxPy is a sample pin

26.4.8. Channel configuration register(TSI_CHCFG)

Address offset: 0x28

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				G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0

rw rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	GxPy	Channel pin mode This bit is set and cleared by software. 0: Pin GxPy is not a channel pin 1: Pin GxPy is a channel pin

26.4.9. Group control register(TSI_GCTL)

Address offset: 0x30

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												GE2	GE1	GE0	

r r r
rw rw rw

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18:16	GCx	Group complete This bit is set by hardware when charge-transfer sequence for an enabled group is complete. It is cleared by hardware when a new charge-transfer sequence starts. 0: Charge-transfer for group x is not complete 1: Charge-transfer for group x is complete
15:3	Reserved	Must be kept at reset value
2:0	GEx	Group enable This bit is set and cleared by software. 0: Group x is disabled 1: Group x is enabled

26.4.10. Group x cycle number registers(TSI_GxCYCN)(x = 0..2)

Address offset: 0x30 + 0x04 *(x + 1)

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															CYCN[13:0]

rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
13:0	CYCN[13:0]	<p>Cycle number</p> <p>These bits reflect the cycle number for a group as soon as a charge-transfer sequence completes. They are cleared by hardware when a new charge-transfer sequence starts.</p>

26.4.11. Control register1 (TSI_CTL1)

Address offset: 0x300

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:28	ECDIV[2:1]	<p>Extend Charge clock(ECCLK) division factor.</p> <p>ECCLK in TSI is divided from HCLK and ECDIV defines the division factor.</p> <p>0x0: $f_{ECCLK} = f_{HCLK}$</p> <p>0x1: $f_{ECCLK} = f_{HCLK}/2$</p> <p>0x2: $f_{ECCLK} = f_{HCLK}/3$</p>

0x3: $f_{ECCLK} = f_{HCLK}/4$

0x4: $f_{ECCLK} = f_{HCLK}/5$

0x5: $f_{ECCLK} = f_{HCLK}/6$

0x6: $f_{ECCLK} = f_{HCLK}/7$

0x7: $f_{ECCLK} = f_{HCLK}/8$

Note: ECDIV[2:1] are located in TSI_CTL1 and ECDIV[0] is located in TSI_CTL0.

27:25 Reserved Must be kept at reset value

24 CTCDIV[3] Charge Transfer clock(CTCLK) division factor.

CTCLK in TSI is divided from HCLK and CTCDIV defines the division factor.

0000: $f_{CTCLK} = f_{HCLK}$

0001: $f_{CTCLK} = f_{HCLK}/2$

0010: $f_{CTCLK} = f_{HCLK}/4$

0011: $f_{CTCLK} = f_{HCLK}/8$

....

0111: $f_{CTCLK} = f_{HCLK}/128$

1000: $f_{CTCLK} = f_{HCLK}/256$

1001: $f_{CTCLK} = f_{HCLK}/512$

....

1110: $f_{CTCLK} = f_{HCLK}/16384$

1111: $f_{CTCLK} = f_{HCLK}/32768$

Note: CTCDIV[3] is located in TSI_CTL1 and CTCDIV[2:0] are located in TSI_CTL0.

23:0 Reserved Must be kept at reset value

27. Cryptographic Acceleration Unit (CAU)

27.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with DES, Triple-DES or AES (128, 192, or 256) algorithms. It is fully compliant implementation of the following standards:

- The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDEA) are announced by Federal Information Processing Standards Publication (FIPS) 46-3, October 25, 1999. It follows the American National Standards Institute (ANSI) X9.52 standard.
- The Advanced Encryption Standard (AES) is announced by Federal Information Processing Standards Publication 197, November 26, 2001.

DES/TDES/AES algorithms with different key sizes are supported to perform data encryption and decryption in the CAU in multiple modes.

The CAU is a 32-bit peripheral, DMA transfer is supported and data can be accessed in the input and output FIFO.

27.2. Characteristics

- DES, TDES and AES encryption/decryption algorithms are supported.
- Multiple modes are supported respectively in DES, TDES and AES, including Electronic codebook (ECB), Cipher block chaining (CBC), Counter mode (CTR), Galois/counter mode (GCM), Galois message authentication code mode (GMAC), Counter with CBC-MAC (CCM), Cipher Feedback mode (CFB) and Output Feedback mode(OFB).
- DMA transfer for incoming and outgoing data is supported.

DES/TDES

- Supports the ECB and CBC chaining algorithms
- Two 32-bit initialization vectors (IV) are used in CBC mode
- 8*32-bit input and output FIFO
- Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping
- Data are transferred by DMA, CPU during interrupts, or without both of them

AES

- Supports the ECB, CBC, CTR, GCM, GMAC, CCM, CFB and OFB chaining algorithms
- Supports 128-bit, 192-bit and 256-bit keys
- Four 32-bit initialization vectors (IV) are used in CBC, CTR, GCM, GMAC, CCM, CFB

- and OFB modes
- 8*32-bit input and output FIFO
 - Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping
 - Data can be transferred by DMA, CPU during interrupts, or without both of them

27.3. CAU data type and initialization vectors

27.3.1. Data type

The cryptographic acceleration unit receives data of 32 bits at a time, while they are processed in 64/128 bits for DES/AES algorithms. For each data block, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the cryptographic acceleration processor. The same swapping operation should be also performed on the processor output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian.

[Figure 27-1. DATAM No swapping and Half-word swapping](#) and [Figure 27-2. DATAM Byte swapping and Bit swapping](#) illustrate the 128-bit AES block data swapping according to different data types. (For DES, the data block is two 32-bit words, please refer to the first two words data swapping in the figure).

Figure 27-1. DATAM No swapping and Half-word swapping

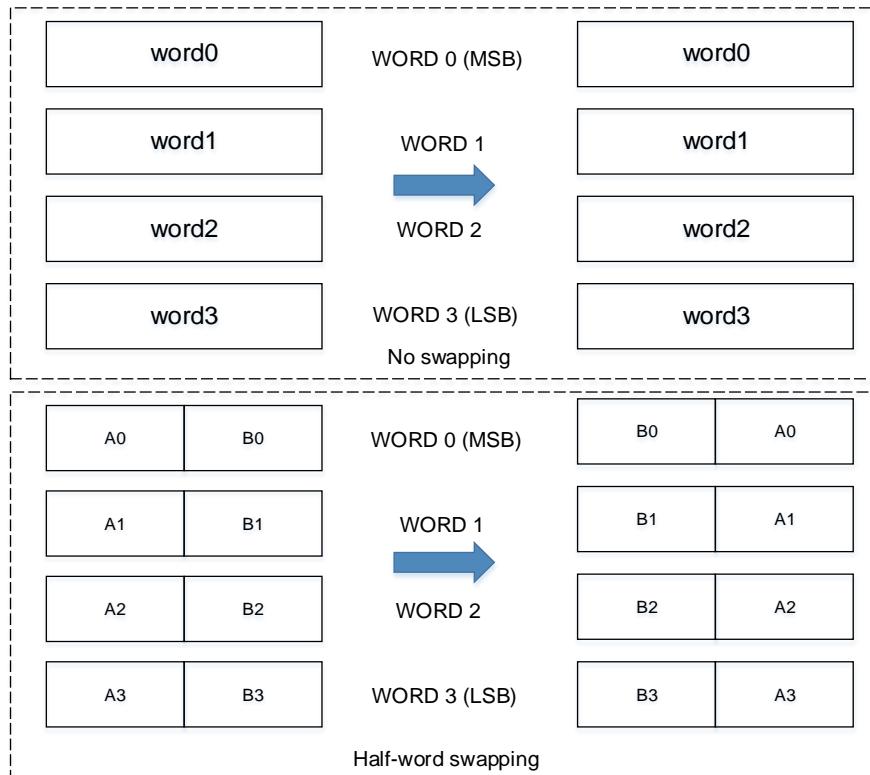
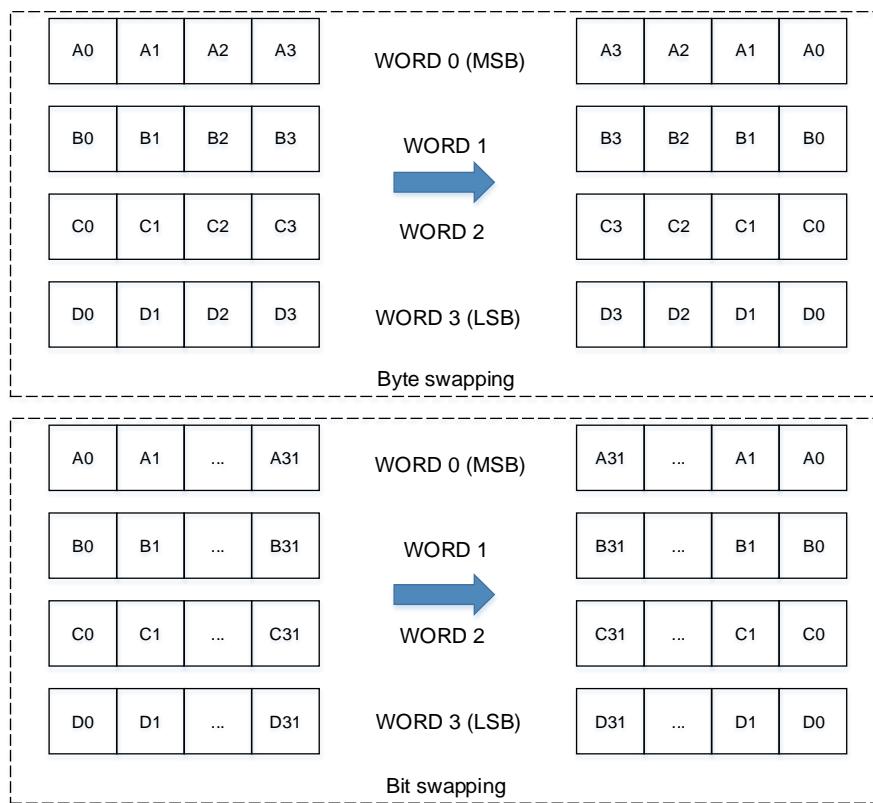


Figure 27-2. DATAM Byte swapping and Bit swapping


27.3.2. Initialization vectors

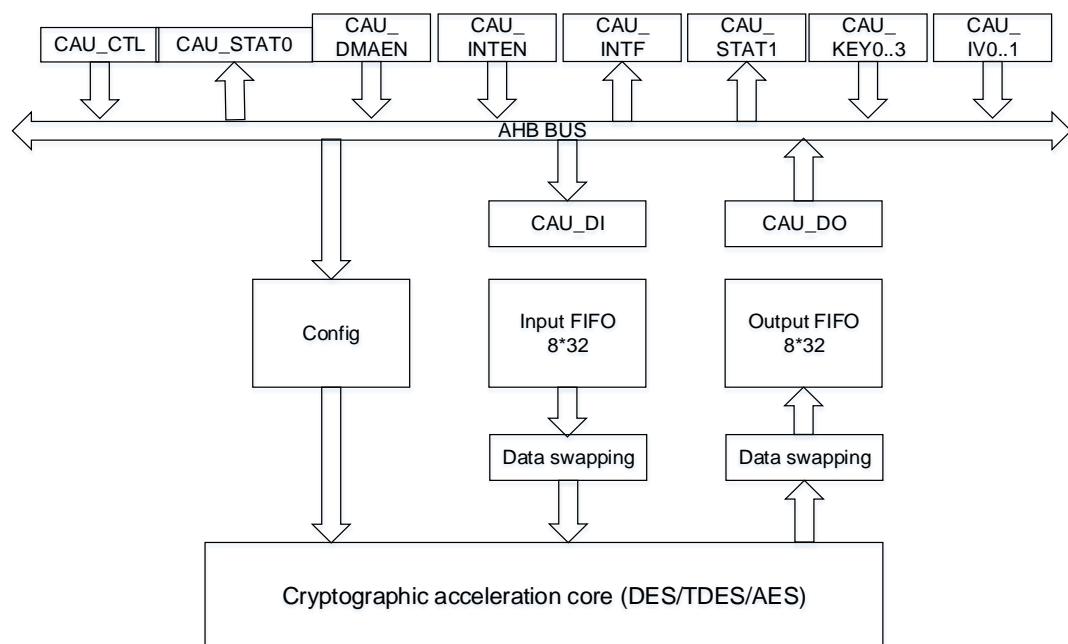
The initialization vectors are used in CBC, CTR, GCM, GMAC, CCM, CFB and OFB modes to XOR with data blocks. They are independent of plaintext and ciphertext, and the DATAM value will not affect them. Note the initialization vector registers CAU_IV0,1(H/L) can only be written when BUSY is 0, otherwise the write operations are invalid.

27.4. Cryptographic acceleration processor

The cryptographic acceleration unit implements DES and AES acceleration processors, which are detailed described in section [DES/TDES cryptographic acceleration processor](#) and [AES cryptographic acceleration processor](#).

[**Figure 27-3. CAU diagram**](#) shows the block diagram of the cryptographic acceleration unit.

Figure 27-3. CAU diagram



27.4.1. DES/TDES cryptographic acceleration processor

The DES/TDES cryptographic acceleration processor contains the DES algorithm (DEA), cryptographic keys (DES algorithm needs 1 key and TDES algorithm needs 3 keys), and initialization vectors in CBC mode.

DES/TDES key

[KEY1] is used in DES and [KEY3 KEY2 KEY1] are used in TDES respectively.

When TDES algorithm is configured, three different keying options are allowed:

1. Three same keys

The three keys KEY3, KEY2 and KEY1 are completely equal, which means KEY3 = KEY2 = KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 -1998) refers to this option. It is easy to understand that this mode is equivalent to DES.

2. Two different keys

In this option, KEY2 is different from KEY1, and KEY3 is equal to KEY1, which means, KEY1 and KEY2 are independent while KEY3 = KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 – 1998) refers to this option.

3. Three different keys

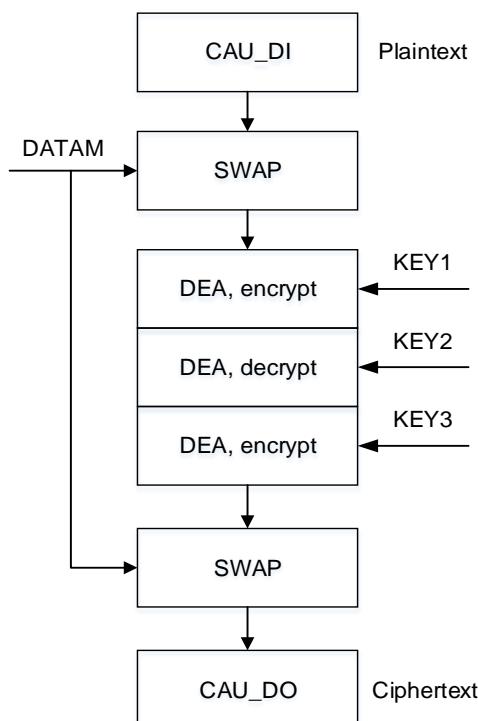
In this option, KEY1, KEY2 and KEY3 are completely independent. FIPS PUB 46-3 -1999 (and ANSI X9.52 – 1998) refers to this option.

More information of the thorough explanation of the key used in the DES/TDES please refer to FIPS PUB 46-3 (and ANSI X9.52 -1998), and the explanation process is omitted in this manual.

DES/TDES ECB encryption

The 64-bit input plaintext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The output after above processes is then swapped back according to the data type again, and a 64-bit ciphertext is produced. When the DES algorithm is configured, the result of the first DEA encrypted using KEY1 is swapped directly according to the data type, and a 64-bit ciphertext is produced. The procedure of DES/TDES ECB mode encryption is illustrated in [Figure 27-4. DES/TDES ECB encryption](#).

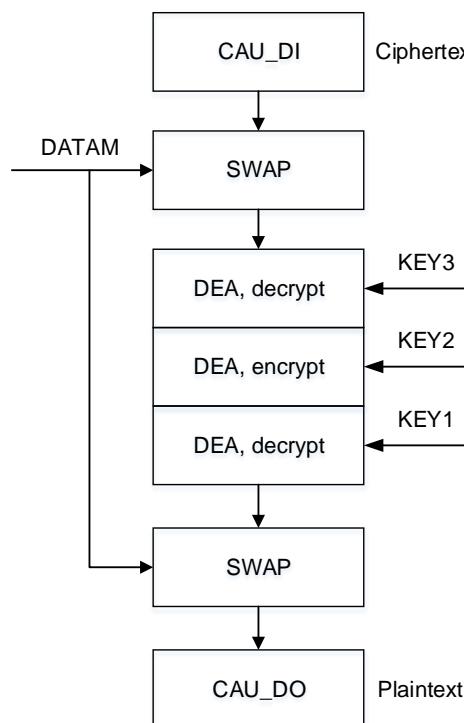
Figure 27-4. DES/TDES ECB encryption



DES/TDES ECB decryption

The 64-bit input ciphertext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The output after above process is then swapped back according to the data type again, and a 64-bit plaintext is produced. When the DES algorithm is configured, the result of the first DEA decrypted using KEY1 is swapped directly according to the data type, and a 64-bit plaintext is produced. The procedure of DES/TDES ECB mode decryption is illustrated in [Figure 27-5. DES/TDES ECB decryption](#).

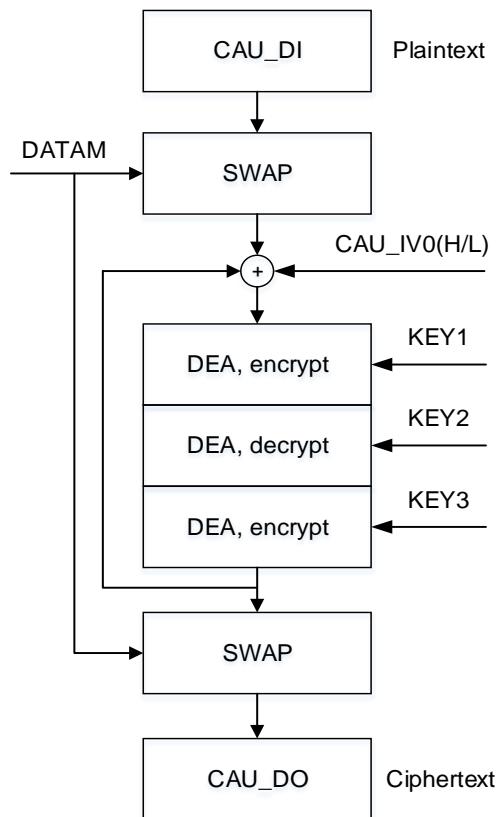
Figure 27-5. DES/TDES ECB decryption



DES/TDES CBC encryption

The input data of the DEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. When the TDES algorithm is configured, the XOR result of the swapped plaintext data block and the 64-bit initialization vector CAU_IV0..1 is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The result is then used as the next initialization vector and exclusive-ORed with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should be omitted. The procedure of DES/TDES CBC mode encryption is illustrated in [Figure 27-6. DES/TDES CBC encryption](#).

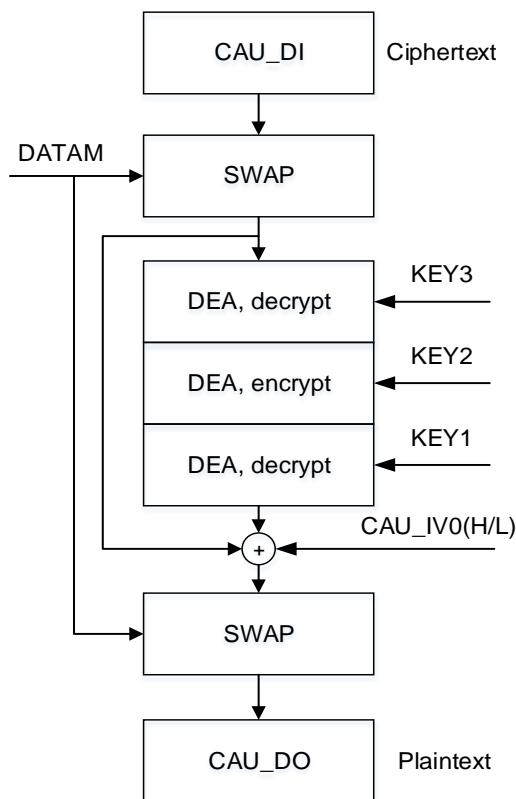
Figure 27-6. DES/TDES CBC encryption



DES/TDES CBC decryption

In DES/TDES CBC decryption, when the TDES algorithm is configured, the first ciphertext block is used directly after data swapping according to the data type, it is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The first result of above process is then XORed with the initialization vector which is the same as that used during encryption. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-ORed with the next result after DEA blocks. The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output plaintext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should also be omitted. The procedure of DES/TDES CBC mode decryption is illustrated in [Figure 27-7. DES/TDES CBC decryption](#).

Figure 27-7. DES/TDES CBC decryption



27.4.2. AES cryptographic acceleration processor

The AES cryptographic acceleration processor consists of three components, including the AES algorithm (AEA), multiple keys and the initialization vectors or Nonce.

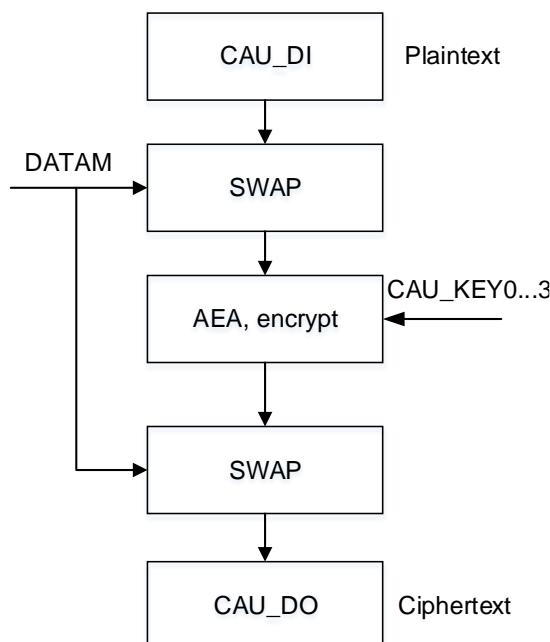
Three lengths of AES keys are supported: 128, 192 and 256 bits, and different initialization vectors or nonce are used depends on the operation mode.

The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

The thorough explanation of the key used in the AES is provided in FIPS PUB 197 (November 26, 2001), and the explanation process is omitted in this manual.

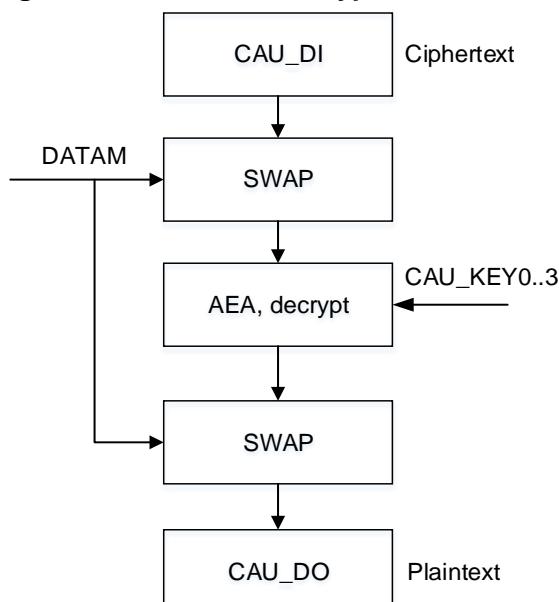
AES-ECB mode encryption

The 128-bit input plaintext is first obtained after data swapping according to the data type. The input data block is read in the AEA and encrypted using the 128, 192 or 256 -bit key. The output after above process is then swapped back according to the data type again, and a 128-bit ciphertext is produced and stored in the out FIFO. The procedure of AES ECB mode encryption is illustrated in [Figure 27-8. AES ECB encryption](#).

Figure 27-8. AES ECB encryption


AES-ECB mode decryption

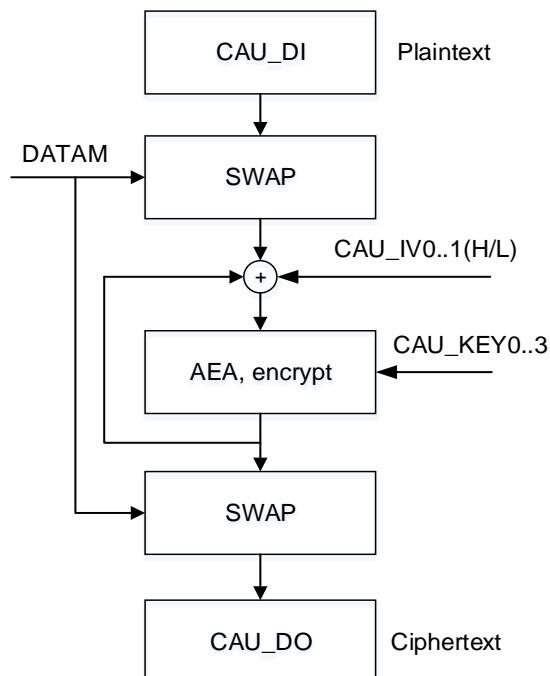
First of all, the key derivation must be completed to prepare the decryption keys, the input key of the key schedule is the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. The output is then swapped back according to the data type again, and a 128-bit plaintext is produced. The procedure of AES ECB mode decryption is illustrated in [Figure 27-9. AES ECB decryption](#).

Figure 27-9. AES ECB decryption


AES-CBC mode encryption

The input data of the AEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. The XOR result of the swapped plaintext data block and the 128-bit initialization vector CAU_IV0..1 is read in the AEA and encrypted using the 128-, 192-, 256-bit key. The result is then used as the next initialization vector and exclusive-ORed with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. The procedure of AES CBC mode encryption is illustrated in [Figure 27-10. AES CBC encryption](#).

Figure 27-10. AES CBC encryption

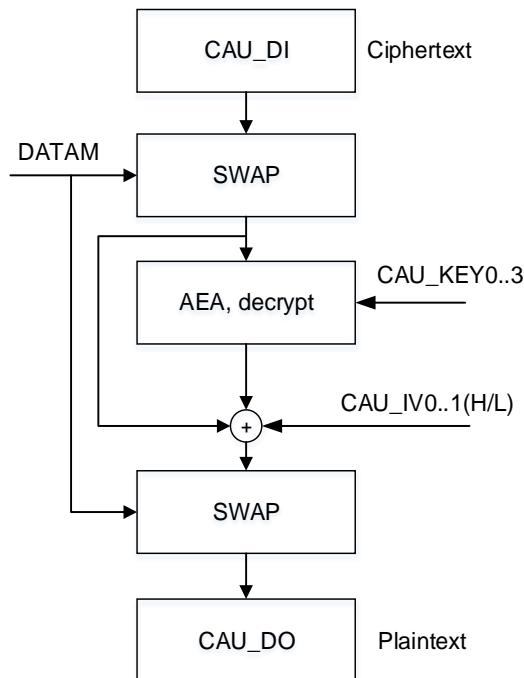


AES-CBC mode decryption

Similar to that in AES-ECB mode decryption, the key derivation also must be completed first to prepare the decryption keys, the input of the key schedule should be the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-ORed with the next result after AEA blocks (The first initialization is obtained directly from the CAU_IV0..1 registers). The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output

plaintext is also obtained after data swapping according to the data type. The procedure of AES CBC mode decryption is illustrated in [Figure 27-11. AES CBC decryption](#).

Figure 27-11. AES CBC decryption



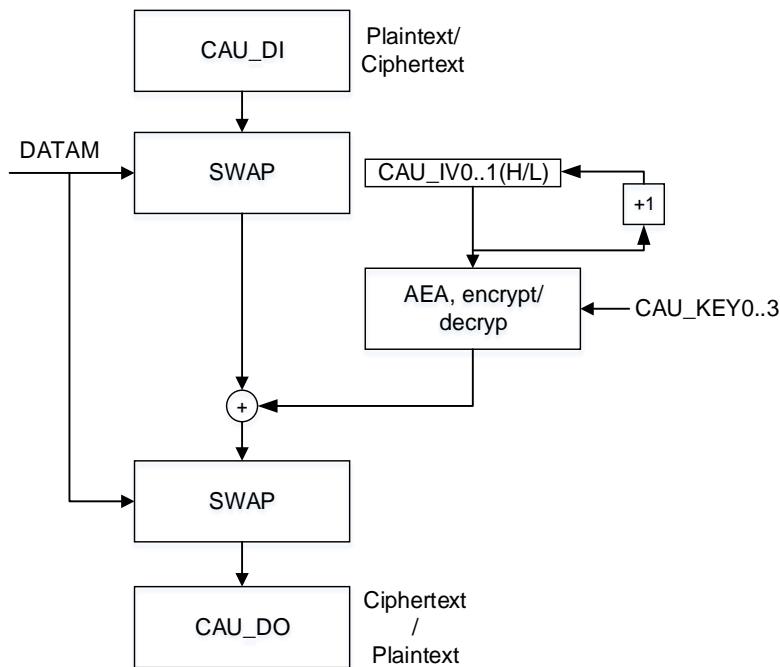
AES-CTR mode

In counter mode, a counter is used in addition with a nonce value to be encrypted and decrypted in AEA, and the result will be used for the XOR operation with the plaintext or the ciphertext. As the counter is incremented from the same initialized value for each block in encryption and decryption, the key schedules during the encryption and decryption are the same. Then decryption operation acts exactly in the same way as the encryption operation. Only the 32-bit LSB of the 128-bit initialization vector represents the counter, which means the other 96 bits are unchanged during the operation, and the initial value should be set to 1. Nonce is 32-bit single-use random value and should be updated to each communication block. And the 64-bit initialization vector is used to ensure that a given value is used only once for a given key. [Figure 27-12. Counter block structure](#) illustrates the counter block structure and [Figure 27-13. AES CTR encryption/decryption](#) shows the AES CTR encryption/decryption.

Figure 27-12. Counter block structure

NONCE	Initialization vector	Counter
-------	-----------------------	---------

Figure 27-13. AES CTR encryption/decryption



AES-GCM mode

The AES Galois/counter mode (GCM) can be used to encrypt or authenticate message, then ciphertext and tag can be obtained. This algorithm is based on AES CTR mode to ensure confidentiality. A multiplier over a fixed finite field is used to generate the tag.

In this mode, four steps are required to perform an encryption/decryption:

1. GCM prepare phase

The hash key is calculated and saved internally to be used later.

- (a) Clear the CAUEN bit to make sure CAU is disabled.
- (b) Configure the ALGM[3:0] bits to '1000'.
- (c) Configure GCM_CCMPH[1:0] bits to '00'.
- (d) Configure key registers and initialization vectors.
- (e) Enable CAU by writing 1 to CAUEN bit.
- (f) Wait until CAUEN bit is cleared by hardware, and then enable CAU again for following phases.

2. GCM AAD (additional authenticated data) phase

This phase must be performed after GCM prepare phase and also precede the encryption/decryption phase. In this phase, data is authenticated but not protected.

- (g) Configure GCM_CCMPH[1:0] bits to '01'.
- (h) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the AAD must be a multiple of 128bits. DMA can also be used.

- (i) Repeat (h) until all AAD data are supplied, wait until BUSY bit is cleared.

3. GCM encryption/decryption phase

This phase must be performed after GCM AAD phase. In this phase, the message is authenticated and encrypted/decrypted.

- (j) Configure GCM_CCMPH[1:0] bits to '10'.
- (k) Configure the computation direction in CAUDIR.
- (l) Write payload data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If the output FIFO is not empty, read the CAU_DO register. DMA can also be used.
- (m) Repeat (l) step until all payload blocks are processed.

4. GCM tag phase

In this phase, the final authentication tag is generated.

- (n) Configure GCM_CCMPH[1:0] bits to '11'.
- (o) Write the input into the CAU_DI register, 4 times write operation is needed. The input consists of the AAD size (64bits) and the payload data size (64bits).
- (p) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.
- (q) Disable the CAU.

Note: The key should be prepared at the beginning when a decryption is performed.

AES-GMAC mode

The AES Galois message authentication code mode is also supported to authenticate the message. It is processing based on the AES-GCM mode, while the encryption/decryption phase is by-passed.

AES-CCM mode

The AES combined cipher machine mode, which is similar to AES-GCM mode, also allows encrypting and authenticating message. It is also based on AES-CTR mode to ensure confidentiality. In this mode, AES-CBC is used to generate a 128-bit tag.

The CCM standard (RFC 3610 Counter with CBC-MAC (CCM) dated September 2003) defines particular encoding rules for the first authentication block (B0 in the standard). In particular, the first block includes flags, a nonce and the payload length expressed in bytes. The CCM standard also specifies another format for encryption/decryption, called A or counter. The counter is incremented during the encryption/decryption phase and its 32 LSB bits are initialized to '1' during the tag generation (A0 packet in the CCM standard).

Note: The formatting operation of B0 packet should be handled by software.

In this mode, four steps are required to perform an encryption/decryption:

1. CCM prepare phase

In this phase, B0 packet (the first packet) is programmed into the CAU_DI register. CAU_DO never contain data in this phase.

- (a) Clear the CAUEN bit to make sure CAU is disabled.
- (b) Configure the ALGM[3:0] bits to '1001'.
- (c) Configure GCM_CCMPH[1:0] bits to '00'.
- (d) Configure key registers and initialization vectors.
- (e) Enable CAU by writing 1 to CAUEN bit.
- (f) Program the B0 packet into the CAU_DI.
- (g) Wait until CAUEN is cleared by hardware, and then enable CAU again for following phases.

2. CCM AAD (additional authenticated data) phase

This phase must be performed after CCM prepare phase and also precede the encryption/decryption phase. In this phase, CAU_DO never contain data in this phase.

This phase can be by-passed if there is no additional authenticated data.

- (h) Configure GCM_CCMPH[1:0] bits to '01'
- (i) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the AAD must be a multiple of 128 bits. DMA can also be used.
- (j) Repeat (i) until all AAD data are supplied, wait until BUSY bit is cleared

3. CCM encryption/decryption phase

This phase must be performed after CCM AAD phase. In this phase, the message is authenticated and encrypted/decrypted.

Like GCM, the CCM chaining mode can be applied on a message composed only by plaintext authenticated data (that is, only AAD, no payload). Note that this way of using CCM is not called CMAC (it is not similar to GCM/GMAC).

- (k) Configure GCM_CCMPH[1:0] bits to '10'
- (l) Configure the computation direction in CAUDIR
- (m) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If so, read the CAU_DO register. DMA can also be used.
- (n) Repeat (m) step until all payload blocks are processed.

4. CCM tag phase

In this phase, the final authentication tag is generated.

- (o) Configure GCM_CCMPH[1:0] bits to '11'
- (p) Write the 128 bit input into the CAU_DI register, 4 times of write operation to CAU_DI is needed. The input is the A0 value.

-
- (q) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.
 - (r) Disable the CAU

AES-CFB mode

The Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and the decryption process is similar to the encryption described before.

AES-OFB mode

The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an initialization vectors (IV) to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and the decryption process is similar to the encryption described before.

27.5. Operating modes

Encryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H/L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the algorithm (DES/TDES/AES) and the chaining mode (ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
6. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
7. Configure the initialization vectors by writing the CAU_IV0..1 registers.
8. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
9. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
10. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA/CPU during interrupts/no DMA or interrupts.
11. Wait for ONE bit in the CAU_STAT0 register is 1 then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
12. Repeat steps 10, 11 until all data blocks has been encrypted.

Decryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.

2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H/L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the ALGM[3:0] bits to “0111” in the CAU_CTL register to complete the key derivation.
6. Enable the CAU by set the CAUEN bit as 1.
7. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
8. Configure the algorithm (DES/TDES/AES) and the chaining mode (ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
9. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
10. Configure the initialization vectors by writing the CAU_IV0..1 registers.
11. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
12. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
13. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA/CPU during interrupts/no DMA or interrupts.
14. Wait for ONE bit in the CAU_STAT0 register is 1, then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
15. Repeat steps 13, 14 until all data blocks has been decrypted.

27.6. CAU DMA interface

The DMA can be used to transfer data blocks with the interface of the cryptographic acceleration unit. The operations can be controlled by the CAU_DMAEN register. DMAIEN is used to enable the DMA request during the input phase, then a word is written into CAU_DI from DMA. DMAOEN is used to enable the DMA request during the output phase, then a word is read from the CAU.

Single and Burst transfers are both supported to ensure the data transfer if the number of words is not an integral multiple of burst size. Note the DMA controller should be configured to perform burst of 4 words or less to make sure no data will be lost. DMA channel for output data has a higher priority than that channel for input data so that the output FIFO can be empty earlier than that the input FIFO is full.

27.7. CAU interrupts

There are two types of interrupt registers in CAU, which are CAU_STAT1 and CAU_INTF. In CAU, the interrupt is used to indicate the situation of the input and output FIFO.

Any of input and output FIFO interrupt can be enabled or disabled by configuring the Interrupt

Enable register CAU_INTEN. Value 1 of the register enable the interrupts.

Input FIFO interrupt

The input FIFO interrupt is asserted when the number of words in the input FIFO is less than four words, then ISTA is asserted. And if the input FIFO interrupt is enabled by IINTEN with a 1 value, the IINTF is also asserted. Note if the CAUEN is low, then the ISTA and IINTF are also always low.

Output FIFO interrupt

The output FIFO interrupt is asserted when the number of words in the output FIFO is more than one words, then OSTA is asserted. And if the output FIFO interrupt is enabled by OINTEN with a 1 value, the OINTF is also asserted. Note Unlike that of Input FIFO interrupt, the value of CAUEN will never affect the situation of OSTA and OINTF.

27.8. CAU suspended mode

It is possible to suspend a data block if another new data block with a higher priority needs to be processed in CAU. The following steps can be performed to complete the encryption/decryption acceleration of the suspended data blocks.

When DMA transfer is used:

1. Stop the current input transfer. Clear the DMAIEN bit in the CAU_DMAEN register.
2. When it is DES or AES, wait until both the input and output FIFO are both empty if the input FIFO is not empty (IEM = 0), then write a word of data into CAU_DI register, do as so until the IEM is checked to be 1, then wait until the BUSY bit is cleared, so that the next data block will not be affected by the last one. Case of TDES is similar to that of AES except that it does not need to wait until the input FIFO is empty.
3. Stop the output transfer by clearing the DMAOEN bit in the CAU_DMAEN register. And disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
4. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC, or CCM mode, the context switch registers CAU_GCMCCMCTXS_x ($x = 0..7$) and CAU_GCMCTXS_x ($x = 0..7$) should also be stored.
5. Configure and process the new data block.
6. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors, and the context switch registers CAU_GCMCCMCTXS_x ($x = 0..7$) and CAU_GCMCTXS_x ($x = 0..7$) should also be restored. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

When data transfer is done by CPU access to CAU_DI and CAU_DO:

1. When the data transfer is done by CPU access, then wait for the fourth read of the CAU_DO register and before the next CAU_DI write access so that the message is suspended at the end of a block processing.
2. Disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
3. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC, or CCM mode, the context switch registers CAU_GCMCCMCTXS x ($x = 0..7$) and CAU_GCMCTXS x ($x = 0..7$) should also be stored.
4. Configure and process the new data block.
5. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors, and the context switch registers CAU_GCMCCMCTXS x ($x = 0..7$) and CAU_GCMCTXS x ($x = 0..7$) should also be restored. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

27.9. Register definition

CAU secure access base address: 0x5C06 0000
 CAU non-secure access base address: 0x4C06 0000

27.9.1. Control register (CAU_CTL)

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										ALGM[3]	Reserved	GCM_CCMPPH[1:0]			
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUEN	FFLUSH	Reserved			KEYM[1:0]		DATAM[1:0]		ALGM[2:0]		CAUDIR	Reserved			
rw	w				rw		rw		rw		rw		rw		rw

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	ALGM[3]	Encryption/decryption algorithm mode bit 3
18	Reserved	Must be kept at reset value.
17:16	GCM_CCMPPH[1:0]	GCM CCM phase 00: prepare phase 01: AAD phase 10: encryption/decryption phase 11: tag phase
15	CAUEN	CAU Enable 0: CAU is disabled 1: CAU is enabled Note: the CAUEN can be cleared automatically when the key derivation (ALGM = 0111b) is finished or the AES-GCM or AES-CCM prepare phase finished.
14	FFLUSH	Flush FIFO 0: No effect 1: When CAUEN = 1, flush the input and output FIFO Reading this bit always returns 0
13:10	Reserved	Must be kept at reset value.
9:8	KEYM[1:0]	AES key size mode configuration, must be configured when BUSY = 0 00: 128-bit key length

		01: 192-bit key length 10: 256-bit key length 11: never use
7:6	DATAM[1:0]	Data swapping type mode configuration, must be configured when BUSY = 0 00: No swapping 01: Half-word swapping 10: Byte swapping 11: Bit swapping
5:3	ALGM[2:0]	Encryption/decryption algorithm mode bit 0 to bit 2 These bits and bit 19 of CAU_CTL must be configured when BUSY = 0 0000: TDES-ECB with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0001: TDES-CBC with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0) is used to XOR with data blocks 0010: DES-ECB with only CAU_KEY1 Initialization vectors (CAU_IV0..1) are not used 0011: DES-CBC with only CAU_KEY1 Initialization vectors (CAU_IV0) is used to XOR with data blocks 0100: AES-ECB with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0101: AES-CBC with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks 0110: AES_CTR with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks In this mode, encryption and decryption are same, then the CAUDIR is disregarded. 0111: AES key derivation for decryption mode. The input key must be same to that used in encryption. The BUSY bit is set until the process has been finished, and CAUEN is then cleared. 1000: Galois Counter Mode (GCM). This algorithm mode is also used for GMAC algorithm. 1001: Counter with CBC-MAC (CCM). 1010: Cipher Feedback (CFB) mode 1011: Output Feedback (OFB) mode
2	CAUDIR	CAU direction, must be configured when BUSY = 0 0: encryption 1: decryption
1:0	Reserved	Must be kept at reset value.

27.9.2. Status register 0 (CAU_STAT0)

Address offset: 0x04

Reset value: 0x0000 0003

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 BUSY OFU ONE INF IEM

r r r r r r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	BUSY	Busy bit 0: No processing. This is because: - CAU is disabled by CAUEN = 0 or the processing has been completed. - No enough data or no enough space in the input/output FIFO to perform a data block 1: CAU is processing data or key derivation.
3	OFU	Output FIFO is full 0: Output FIFO is not full 1: Output FIFO is full
2	ONE	Output FIFO is not empty 0: Output FIFO is empty 1: Output FIFO is not empty
1	INF	Input FIFO is not full 0: Input FIFO is full 1: Input FIFO is not full
0	IEM	Input FIFO is empty 0: Input FIFO is not empty 1: Input FIFO is empty

27.9.3. Data input register (CAU_DI)

Address offset: 0x08

Reset value: 0x0000 0000

The data input register is used to transfer plaintext or ciphertext blocks into the input FIFO for processing. The MSB is firstly written into the FIFO and the LSB is the last one. If the CAUEN is 0 and the input FIFO is not empty, when it is read, then the first data in the FIFO is popped out and returned. If the CAUEN is 1, the returned value is undefined. Once it is read, then the FIFO must be flushed.

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
DI[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DI[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DI[31:0]	Data input Write these bits will write data to IN FIFO, read these bits will return IN FIFO value if CAUEN is 0, or it will return an undefined value

27.9.4. Data output register (CAU_DO)

Address offset: 0x0C

Reset value: 0x0000 0000

The data output register is a read only register. It is used to receive plaintext or ciphertext results from the output FIFO. Similar to CAU_DI, the MSB is read at first while the LSB is read at last.

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
DO[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO[15:0]															
r															

Bits	Fields	Descriptions
31:0	DO[31:0]	Data output These bits are read only, read these bits return OUT FIFO value.

27.9.5. DMA enable register (CAU_DMAEN)

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															

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMAOEN	DMA output enable 0: DMA for OUT FIFO data is disabled 1: DMA for OUT FIFO data is enabled
0	DMAIEN	DMA input enable 0: DMA for IN FIFO data is disabled 1: DMA for IN FIFO data is enabled

27.9.6. Interrupt enable register (CAU_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															
														OINTEN	IINTEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTEN	OUT FIFO interrupt enable 0: OUT FIFO interrupt is disable 1: OUT FIFO interrupt is enable
0	IINTEN	IN FIFO interrupt enable 0: IN FIFO interrupt is disable 1: IN FIFO interrupt is enable

27.9.7. Status register 1 (CAU_STAT1)

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

r r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OSTA	OUT FIFO interrupt status 0: OUT FIFO interrupt status not pending 1: OUT FIFO interrupt status pending
0	ISTA	IN FIFO interrupt status 0: IN FIFO interrupt not pending 1: IN FIFO interrupt flag pending

27.9.8. Interrupt flag register (CAU_INTF)

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														OINTF	IINTF

r r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTF	OUT FIFO enabled interrupt flag 0: OUT FIFO Interrupt not pending 1: OUT FIFO Interrupt pending
0	IINTF	IN FIFO enabled interrupt flag 0: IN FIFO Interrupt not pending 1: IN FIFO Interrupt pending when CAUEN is 1

27.9.9. Key registers (CAU_KEY0..3(H/L))

Address offset: 0x20 to 0x3C

Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES mode, only CAU_KEY1 is used.

In TDES mode, CAU_KEY1, CAU_KEY2 and CAU_KEY3 are used.

In AES-128 mode, KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[0:63], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[64:127].

In AES-192 mode, KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[0:63], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[64:127], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[128:191].

In AES-256 mode, KEY0H[31:0] || KEY0L[31:0] is used as AES_KEY[0:63], KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[64:127], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[128:191], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[192:255].

NOTE: “||” is a concatenation operator. For example, X|| Y denotes the concatenation of two bit strings X and Y.

CAU_KEY0H

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0L[15:0]															
w															

CAU_KEY1H

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															
w															

CAU_KEY1L

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															
w															

CAU_KEY2H

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															
w															

CAU_KEY2L

Address offset: 0x34

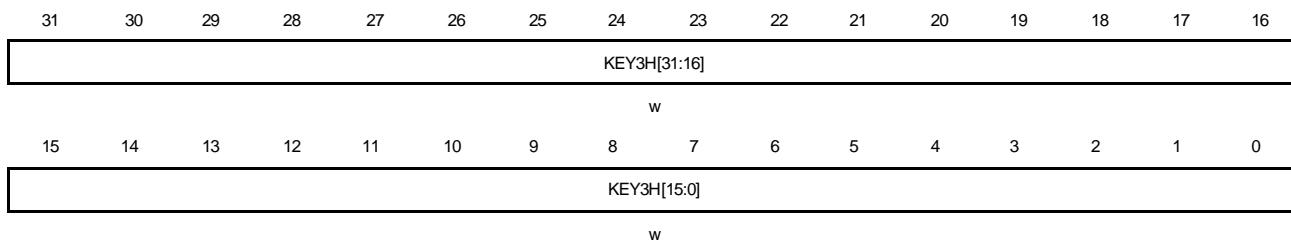
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

Address offset: 0x38

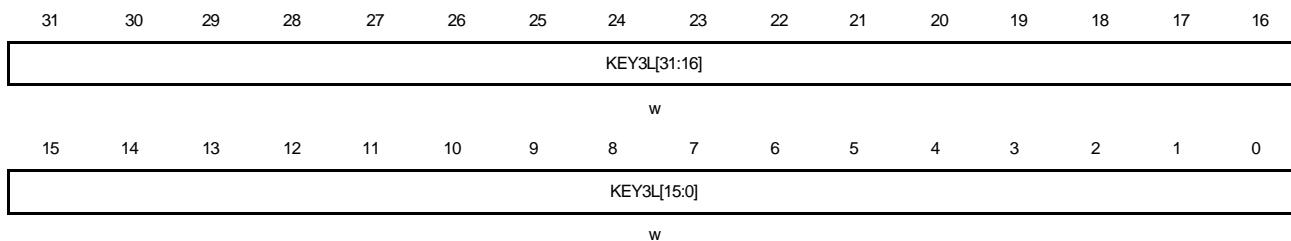
Reset value: 0x0000 0000



CAU_KEY3L

Address offset: 0x3C

Reset value: 0x0000 0000



Bits	Fields	Descriptions
31:0	KEY0...3(H/L)	The key for DES, TDES, AES

27.9.10. Initial vector registers (CAU_IV0..1(H/L))

Address offset: 0x40 to 0x4C

Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES/TDES mode, IV0H is the leftmost bits, and IV0L is the rightmost bits of the initialization vectors.

In AES mode, IV0H is the leftmost bits, and IV1L is the rightmost bits of the initialization vectors.

CAU_IV0H

Address offset: 0x40

Reset value: 0x0000 0000



IV0H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															

CAU_IV0L

Address offset: 0x44

Reset value: 0x0000 0000

IV0L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															

CAU_IV1H

Address offset: 0x48

Reset value: 0x0000 0000

IV1H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															

CAU_IV1L

Address offset: 0x4C

Reset value: 0x0000 0000

IV1L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															

Bits	Fields	Descriptions
31:0	IV0...1(H/L)	The initialization vector for DES, TDES, AES

27.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x = 0..7)

Address offset: 0x50 to 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
CTXx[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. Note: These registers are used only when GCM, GMAC, or CCM mode is selected.

27.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x = 0..7)

Address offset: 0x70 to 0x8C

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
CTXx[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. Note: These registers are used only when GCM or GMAC mode is selected.

28. Hash Acceleration Unit (HAU)

28.1. Overview

The hash acceleration unit is used for information security. The secure hash algorithm (SHA-1, SHA-224, SHA-256), the message-digest algorithm (MD5) and the keyed-hash message authentication code (HMAC) algorithm are supported for various applications. The digest will be computed and the length is 160/224/256/128 bits for a message up to $(2^{64} - 1)$ bits computed by SHA-1, SHA-224, SHA-256 and MD5 algorithms respectively. In HMAC algorithm, SHA-1, SHA-224, SHA-256 or MD5 will be called twice as hash functions and authenticating messages can be produced.

The HAU is fully compliant implementation of the following standards:

- Federal Information Processing Standards Publication 180-2 (FIPS PUB 180-2)
- Secure Hash Standard specifications (SHA-1, SHA-224, SHA-256)
- Internet Engineering Task Force Request for Comments number 1321 (IETF RFC 1321) specifications (MD5)

28.2. Characteristics

- 32-bit AHB slave peripheral
- High performance of computation of hash algorithms
- Little-endian data representation
- Multiple data types are supported, including no swapping, half-word swapping, byte swapping, and bit swapping with 32-bit data words
- Automatic data padding to fill the 512-bit message block for digest computation
- DMA transfer is supported
- Hash/HMAC process suspended mode

28.3. HAU data type

The hash acceleration unit receives data words of 32 bits at a time, while they are processed in 512-bits blocks. For each input word, according to the data type, the data could be bit/byte/half-word/no swapped before they are transferred into the hash acceleration core. The same swapping operation should be also performed on the core output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian. However, the computation of SHA-1, SHA-224 and SHA-256 are big-endian.

[Figure 28-1. DATAM No swapping and Half-word swapping](#) and [Figure 28-2. DATAM Byte swapping and Bit swapping](#) illustrate the data swapping according to different data

types.

Figure 28-1. DATAM No swapping and Half-word swapping

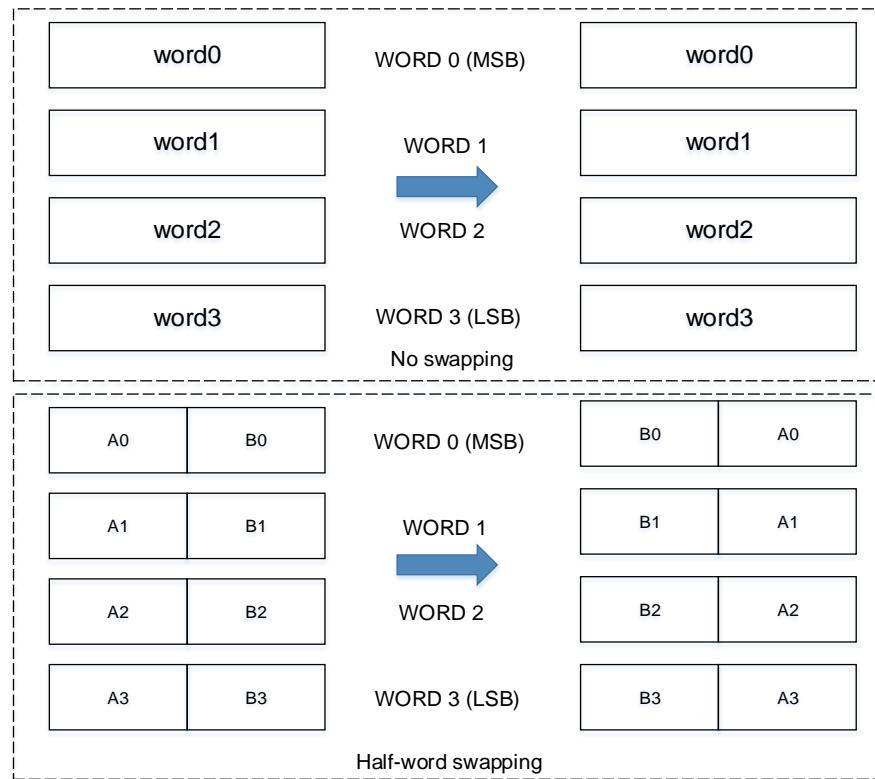
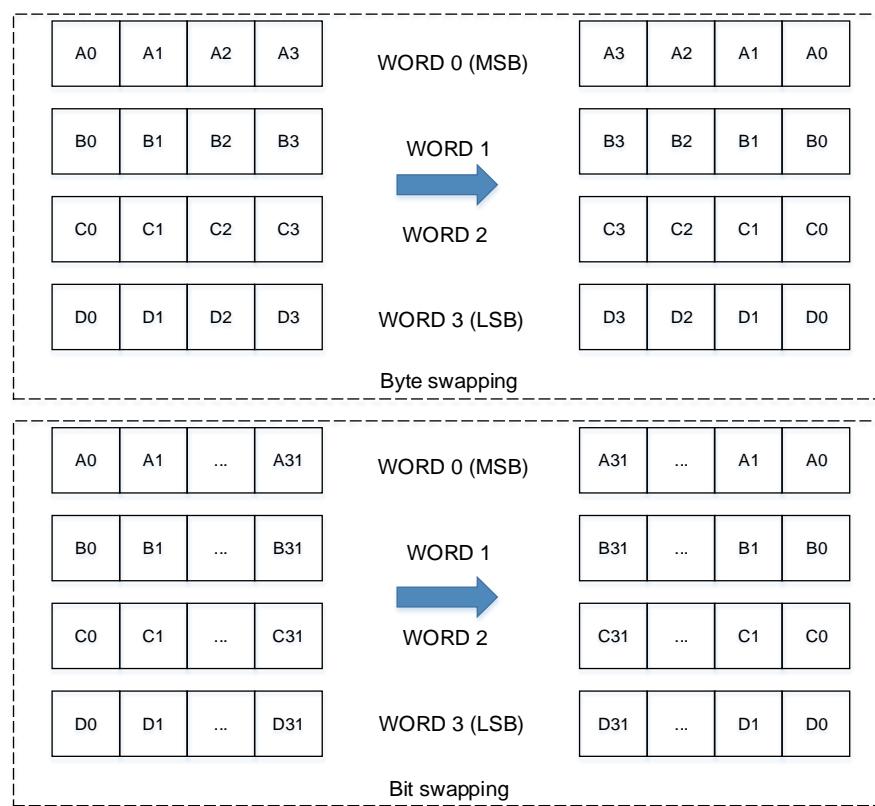


Figure 28-2. DATAM Byte swapping and Bit swapping

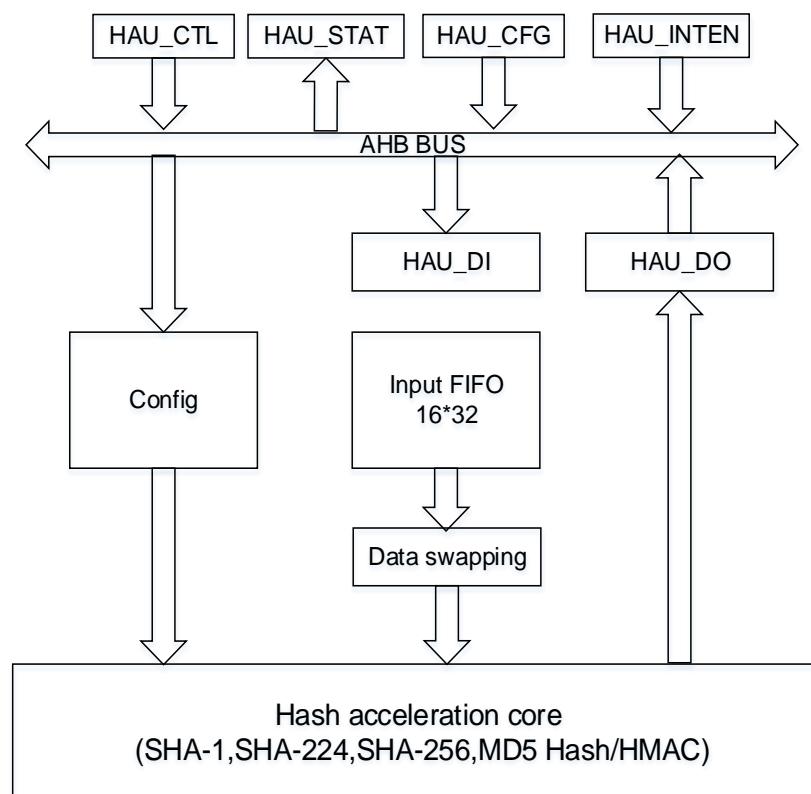


28.4. HAU core

The hash acceleration unit is used to compute condensed information of input messages with secure hash algorithms. The digest result has a length of 160/224/256/128 bits for a message up to $(2^{64}-1)$ bits computed by SHA-1, SHA-224, SHA256 and MD5 algorithms respectively. It can be used to generate or verify the signature of a message with a higher efficiency because of the much simpler of the information.

A message which need to be processed in the HAU should be considered as bit information. And the length is the number of bits of the message. The information security is ensured because that, to find the original message using the digest is computationally impossible and, the result will be completely different with any change to the input message.

Figure 28-3. HAU block diagram



28.4.1. Automatic data padding

The input message should be padded first so that the number of bits in the input of the HAU core can be an integral multiple of 512. First of all, a “1” is added to follow the last bit of the input message, and then several “0” should be padded to ensure the result modulo 512 is 448, at last, a 64-bit length information of input is added.

After the message padding is correctly performed, the VBL bits in the HAU_CFG register is configured as the 64-bit length value above, and CALEN bit in the HAU_CFG register can be

set 1 to start the calculation of the digest of the last block.

Data Padding Example: The input message is “HAU”, which ASCII hexadecimal code is:

484155

Then the VBL bits in the HAU_CFG register is set as decimal 24 because of the valid bit length. A “1” is added at bit location 24 then, and several “0” are padded so that the result modulo 512 is 448, the hexadecimal result is as follows:

```
48415580 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000
```

After that, a 64-bit length information of the input message is padded, which hexadecimal value is 18, and the final result will be:

```
48415580 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000018
```

28.4.2. Digest computing

After data padding, for each block calculation of HAU, 512 bits are written into the HAU core by DMA or CPU. To start the processing of the HAU core, the peripheral must obtain the information as to whether the HAU_DI register contains the last bits of the message or not. This can be confirmed with the status of the input FIFO and the HAU_DI register.

When DMA is used to transfer data:

The status of the block transfer is automatically interpreted with the information from the DMA controller. And padding and digest computation are performed automatically as if CALEN bit in the HAU_CFG register is set as 1.

Note: If hash message is large files and multiple DMA transfers are needed, then MDS bit should be set as 1. And the VBL bits need to be set before the transfer. The CALEN bit is not set automatically after an intermediate DMA transfer completed. Only when the last DMA transfer is processing, the MDS bit is cleared so that the CALEN bit is automatically set after data transferring.

Otherwise, the MDS bit is set as 0. And the CALEN bit is set automatically after a DMA transfer. Also, VBL bits need to set before the DMA transfer.

When CPU is used to transfer data without DMA:

- The intermediate block computing can be started when HAU_DI is filled with another new word of the next block
- The last block computing can be started when CALEN bit in the HAU_CFG register is 1.

28.4.3. Hash mode

The hash mode is selected when the HMS bit in the HAU_CTL register is set as 0. And when the START bit in the HAU_CTL register is 1, SHA-1, SHA-224, SHA-256 and MD5 mode computation is chosen by the ALGM bits.

After a message block of 512 bit has been received through the HAU_DI register and the input FIFO, the processor starts the calculation with the information from DMA or the status of the CALEN bit.

The results can be finally read from the HAU_DO0..7 registers.

28.4.4. HMAC mode

HMAC mode is used for message authentication with a unique key chosen by the user. More information about the HMAC specifications please refer to “HMAC: keyed-hashing for message authentication, H. Krawczyk, M. Bellare, R. Canetti, February 1997”.

The HMAC algorithm can be represented as:

$$\text{HMAC}(\text{input}) = \text{HASH}[(\text{key} \mid \text{opad}) \text{ XOR } 0x5c] \mid \text{HASH}[(\text{key} \mid \text{ipad}) \text{ XOR } 0x36] \mid \text{input}]$$

where ipad and opad are used to extend the key to 512 bits with several “0” and | is the concatenation operator.

There are four different phases in the HMAC mode:

1. Configure the HMS bit in the HAU_CTL register as 1 and set the ALGM bits as the desired algorithm. If the key size is longer than 64 bytes, then the KLM bit in the HAU_CTL register should also be set. After that, start the HAU core by set the START bit.
2. The key is used as the input message to complete the calculation in HASH mode.
3. The new key used for the inner hash function is elaborated when the last word is accessed and computation has started.
4. After the first hash round, HAU core starts to receive the key for the outer hash function, usually, the outer hash function uses the same new key as the inner hash function. And when the last word of the key is entered and computation starts, the results are available in the HAU_DO registers.

28.5. HAU suspended mode

It is possible to suspend HASH or HMAC operation to perform a high-prior task first, then after the high-prior task is finished, resume the suspended operation.

When suspending the current task, it is necessary to save the context of the current task from registers to memory, and then the task can be resumed by restoring the context from memory to the HAU registers.

The following steps can be performed to complete the HAU process of the suspended data blocks.

28.5.1. Transfer data by CPU

1. Stop the current data transmission and calculation. Wait for BUSY = 0, and wait for DIF = 1 when NWIF[3:0] is larger than 0 (do not wait for DIF = 1 when NWIF[3:0] is 0). Only when no data block is processing, users can save context.
2. Save the configuration. Save the content of HAU_INTEN, HAU_CFG, HAU_CTL, HAU_CTXS0 to HAU_CTXS37 (HAU_CTXS0 to HAU_CTXS53 when HMAC operation is processing) registers to memory.
3. Configure and process the new message.
4. Restore the process before. Restore the content from memory to HAU_INTEN, HAU_CFG and HAU_CTL registers.
5. Resume the message calculation. Set START bit of HAU_CTL register to 1, to restart a new message digest calculation.
6. Resume the previous core state. Restore the content from memory to HAU_CTXS0 ~ HAU_CTXS37 (HAU_CTXS0 ~ HAU_CTXS53 when HMAC operation is to be resumed) registers.
7. Continue the operation from where it suspended before.

28.5.2. Transfer data by DMA

1. Wait for BUSY = 0, then if the CCF bit of HAU_STAT register is set, the proceeding context switch is no longer need, otherwise wait for BUSY = 1 again.
2. Stop the current data transfer. Disable DMA1 channel 7 data transmission, then clear DMAE bit of HAU_CTL register to disable DMA request.
3. Save the current configuration. Wait for BUSY = 0, then if the CCF bit of HAU_STAT register is set, the proceeding context switch is no longer need, otherwise save the content of HAU_INTEN, HAU_CFG, HAU_CTL, HAU_CTXS0 to HAU_CTXS37 (HAU_CTXS0 to HAU_CTXS53 when HMAC operation is processing) registers to memory.

4. Configure and process the new message.
5. Restore the process before. Restore the content from memory to HAU_INTEN, HAU_CFG and HAU_CTL registers.
6. Resume DMA channel transmission. Reconfigure the DMA channel to transfer data.
7. Resume the message calculation. Set START bit of HAU_CTL register to 1, to restart a new message digest calculation.
8. Resume the previous core state. Restore the content from memory to HAU_CTXS0 ~ HAU_CTXS37 (HAU_CTXS0 ~ HAU_CTXS53 when HMAC operation is to be resumed) registers.
9. Set DMAE bit of HAU_CTL register to 1, continue the operation from where it suspended before.

Note: If the value of NWIF[3:0] bits of HAU_CTL register is 0, it means the context switch occurs between two data blocks, at the time when the previous block is completely processed, and the next block has not been pushed into input FIFO, so there is no need to save and restore HAU_CTXS22 ~ HAU_CTXS37 registers.

28.6. HAU interrupt

There are two types of interrupt registers in HAU, which are both in HAU_STAT register. In HAU, the interrupt is used to indicate the situation of the input FIFO and the status of whether the digest calculation is completed.

Any of interrupts can be enabled or disabled by configuring the HAU interrupt enable register HAU_INTEN. Value 1 of the register bits enable the interrupts.

28.6.1. Input FIFO interrupt

When the processing of data pushed in the input FIFO is completed, then DIF is asserted. If input FIFO interrupt is enabled, when DIF is asserted, input FIFO interrupt will be asserted.

28.6.2. Calculation completion interrupt

When the digest calculation is finished, then CCF is asserted. If calculation completion interrupt is enabled, when CCF is asserted, calculation completion interrupt will be asserted.

28.7. Register definition

HAU secure access base address: 0x5C06 0400

HAU non-secure access base address: 0x4C06 0400

28.7.1. HAU control register (HAU_CTL)

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														ALGM[1]	Reserved
rw														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MDS	DINE		NWIF[3:0]	ALGM[0]	HMS	DATAM[1:0]	DMAE	START					Reserved	
	rw	r		r		rw	rw	rw	rw	rw	rw	w			

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	ALGM[1]	Algorithm selection bit 1
17	Reserved	Must be kept at reset value.
16	KLM	Key length mode 0: Key length \leq 64 bytes 1: Key length $>$ 64 bytes Note: This bit must be changed when no computation is processing.
15:14	Reserved	Must be kept at reset value.
13	MDS	Multiple DMA Selection Set this bit if hash message is large files and multiple DMA transfers are needed. 0: Single DMA transfers needed and CALEN bit is automatically set at the end of a DMA transfer 1: Multiple DMA transfers needed and CALEN bit is not automatically set at the end of a DMA transfer
12	DINE	DI register not empty 0: The DI register is empty 1: The DI register is not empty Note: This bit is cleared when START bit or CALEN bit is set as 1.
11:8	NWIF[3:0]	Number of words in the input FIFO Note: These bits are cleared when START bit set or a digest calculation starts

(CALEN bit is set as 1, or DMA end of transfer)

7	ALGM[0]	Algorithm selection bit 0 This bit and bit 18 of CTL are written by software to select the SHA-1, SHA-224, SHA256 or the MD5 algorithm: 00: Select SHA-1 algorithm 01: Select MD5 algorithm 10: Select SHA224 algorithm 11: Select SHA256 algorithm
6	HMS	HAU mode selection, must be changed when no computation is processing 0: HASH mode selected 1: HMAC mode selected. If the key length is longer than 64 bytes, then KLM bit must also be set
5:4	DATAM[1:0]	Data type mode Defines the format of the data entered into the HAU_DI register: 00: No swapping. The data written to HAU_DI is direct write to FIFO without swapping. 01: Half-word swapping. The data written into HAU_DI need half-word swapping before write to FIFO. 10: Bytes swapping. The data written into HAU_DI need bytes swapping before write to FIFO. 11: Bit swapping. The data written into HAU_DI need bytes swapping before write to FIFO.
3	DMAE	DMA enable 0: DMA disabled 1: DMA enabled Note: 1. This bit is cleared when transferring the last data of the message, but not cleared because of START. 2. When DMA is transferring, writing 0 to this bit will not stop the current transfer until the transfer is completed or START is set as 1.
2	START	Start the digest calculation 1: Start the digest of a new message 0: No effect Note: Reading this bit always returns 0.
1:0	Reserved	Must be kept at reset value.

28.7.2. HAU data input register (HAU_DI)

Address offset: 0x04

Reset value: 0x0000 0000

The data input register is used to transfer message with 512-bit blocks into the input FIFO for

processing. Any new write operation to this register will be extended while the digest calculation is in process until it has been finished.

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
Dl[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dl[15:0]															
rw															

Bits	Fields	Descriptions
31:0	Dl[31:0]	Message data input When write to these registers, the current content pushed to IN FIFO and new value updates. When read, returns the current content.

28.7.3. HAU configuration register (HAU_CFG)

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

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CALEN	Digest calculation enable 0: No calculation 1: Start data padding with VBL prepared previously. Start the calculation of the last digest Note: Reading this bit always returns 0.
7:5	Reserved	Must be kept at reset value.
4:0	VBL[4:0]	Valid bits length in the last word 0x00: All 32 bits of the last data written to HAU_DL after data swapping are valid 0x01: Only bit [31] of the last data written to HAU_DL after data swapping are valid 0x02: Only bits [31:30] of the last data written to HAU_DL after data swapping are valid

0x03: Only bits [31:29] of the last data written to HAU_DI after data swapping are valid

...

0x1F: Only bits [31:1] of the last data written to HAU_DI after data swapping are valid

Note: These bits must be configured before setting the CALEN bit.

28.7.4. HAU data output register (HAU_DO0..7)

Reset value: 0x0000 0000

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

The data output registers are read only registers. They are used to receive results from the output FIFO. And they are reset by the START bit. Any read access when calculating will be extended until the calculation is completed.

In SHA-1 mode, HAU_DO0...4 are used.

In MD5 mode, HAU_DO0...3 are used.

In SHA-224 mode, HAU_DO0...6 are used.

In SHA-256 mode, HAU_DO0...7 are used.

HAU_DO0

Address offset: 0x0C and 0x310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO0[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO0[15:0]															
r															

HAU_DO1

Address offset: 0x10 and 0x314

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO1[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO1[15:0]															
r															

HAU_DO2

Address offset: 0x14 and 0x318

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO2[31:16]															
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO2[15:0]															
								r							

HAU_DO3

Address offset: 0x18 and 0x31C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO3[31:16]															
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO3[15:0]															
								r							

HAU_DO4

Address offset: 0x1C and 0x320

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO4[31:16]															
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO4[15:0]															
								r							

HAU_DO5

Address offset: 0x324

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO5[31:16]															
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO5[15:0]															
								r							

HAU_DO6

Address offset: 0x328

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO6[31:16]															
									r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DO6[15:0]

r

HAU_DO7

Address offset: 0x32C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO7[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO7[15:0]															
r															

Bits	Fields	Descriptions
31:0	DO0..7[31:0]	Message digest result of hash algorithm

28.7.5. HAU interrupt enable register (HAU_INTEN)

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CCIE	Calculation completion interrupt enable 0: Calculation completion interrupt is disabled 1: Calculation completion interrupt is enabled
0	DIIE	Data input interrupt enable 0: Data input interrupt is disabled 1: Data input interrupt is enabled

28.7.6. HAU status and flag register (HAU_STAT)

Address offset: 0x24

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

r r rc_w0 rc_w0

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	BUSY	Busy bit 0: No processing 1: Data block is in process
2	DMAS	DMA status 0: DMA is disabled (DMAE = 0) and no transfer is processing 1: DMA is enabled (DMAE = 1) or a transfer is processing
1	CCF	Digest calculation completion flag 0: Digest calculation is not completed 1: Digest calculation is completed
0	DIF	Data input flag 0: A data is written to data input register 1: A data processing is completed (only the data in input FIFO will be processed)

28.7.7. Context switch register x (HAU_CTXSx) (x = 0...53)

Address offset: 0xF8 + 0x04 × x, (x = 0...53)

Reset value: 0x0000 0002 when x = 0, 0x0000 0000 when x = 1 to 53.

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
CTXx[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															

rw

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The complete internal status of the HAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing.

29. Public Key Cryptographic Acceleration Unit (PKCAU)

29.1. Overview

Public key encryption is also called asymmetric encryption, asymmetric encryption algorithms use different keys for encryption and decryption. The Public Key Cryptographic Acceleration Unit (PKCAU) can accelerate RSA (Rivest, Shamir and Adleman), Diffie-Hellmann (DH key exchange) and ECC (elliptic curve cryptography) in GF(p) (Galois domain). These operations are performed in the Montgomery domain to improve computational efficiency.

29.2. Characteristics

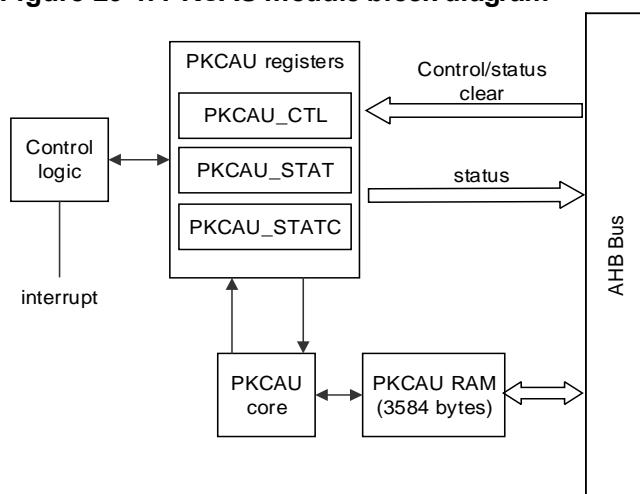
- Support RSA/DH algorithms with up to 3136 bits of operands
- Support ECC algorithm with up to 640 bits of operands
- RSA modular exponentiation, RSA CRT exponentiation
- ECC scalar multiplication, check point on elliptic curve
- ECDSA (Elliptic Curve Digital Signature Algorithm) signature and verification
- Support Montgomery multiplication, accelerate RSA, DH and ECC operations
- Embedded RAM of 3584 bytes
- Conversion between the Montgomery domain and the natural domain
- PKCAU is a 32 bit peripheral, only 32-bit access is supported

29.3. Function overview

The Public Key Cryptographic Acceleration Unit (PKCAU) is used for accelerating RSA, DH and ECC operations in GF(p) domain. The PKCAU module contains PKCAU RAM, PKCAU core, and PKCAU registers. The PKCAU RAM is used to store the parameters required by the operation and save the calculation result after the calculation is completed.

[Figure 29-1. PKCAU module block diagram](#) below provides details on the internal configuration of the PKCAU interface.

Figure 29-1. PKCAU module block diagram



29.3.1. Operands

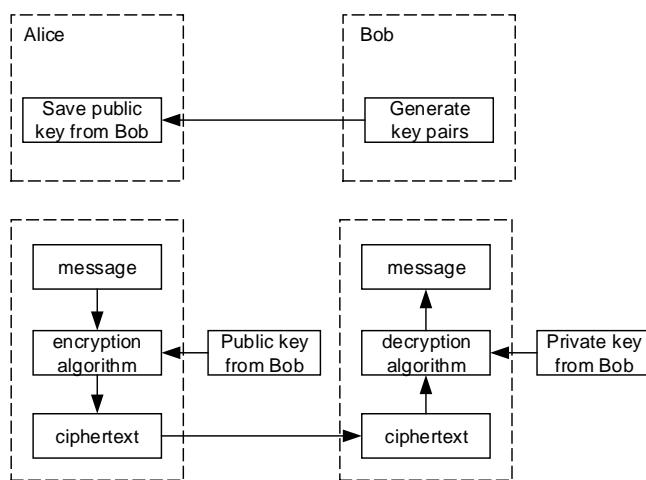
If the RSA operand size is ROS, the modulus length is ML, then the data size is $ROS = (ML/32+1)$ words. If the ECC operand size is EOS, the prime modulus length is ML, then the data size is $EOS = (ML/32+1)$ words.

The PKCAU supports RSA/DH algorithms with up to 3136 bits (98 words) of operands and ECC algorithm with up to 640 bits (20 words) of operands. The maximum ROS is 99 words, and the maximum EOS is 21 words.

When writing the input parameters to the PKCAU RAM, a word 0x00000000 must be added. The PKCAU RAM is little-endian. For example, when writing the input parameter x_p of ECC P256 for ECC scalar multiplication to PKCAU RAM, the modulus length is 8 words, address offset 0x55C stores the lowest byte, address offset 0x578 stores the highest byte. Address offset 0x57C stores word 0x00000000.

29.3.2. RSA algorithm

RSA algorithm is a common public key cryptography algorithm and the most widely used asymmetric cryptography algorithm. The RSA algorithm flow is shown in [Figure 29-2. Flow chart of RSA algorithm](#).

Figure 29-2. Flow chart of RSA algorithm


A complete public key crypto system includes key pairs (public and private keys), encryption algorithms and decryption algorithms.

Generate key pairs of RSA

1. Select two large primes p and q ($p \neq q$);
2. Calculate $n = p \times q$, n is the modulus of public and private keys;
3. Calculate $L = \phi(n) = (p-1)(q-1)$, $\phi(n)$ is euler function;
4. Select e, $1 < e < L$, e and L must be relatively prime;
5. Calculate d, $1 < d < L$ and $e \times d \bmod L = 1$.

The parameters show in [Table 29-1. Parameters of RSA algorithm](#) can be obtained by the above calculation.

Table 29-1. Parameters of RSA algorithm

Parameters	Description
n	modulus
e	public exponent
d	private exponent
(n,e)	public key
(n,d)	private key

RSA encryption

Bob generates a key pair that conforms to RSA algorithm standard, including public key and private key. Bob sends the public key to Alice, and holds the private key. Alice can encrypt the message m through the public key from Bob and obtain the ciphertext c. Then Alice sends the ciphertext to Bob. The ciphertext is $c = m^e \bmod n$.

RSA decryption

After receiving the ciphertext, Bob decrypts the ciphertext to get the plaintext by the private key. The decryption process is $m = c^d \bmod n$.

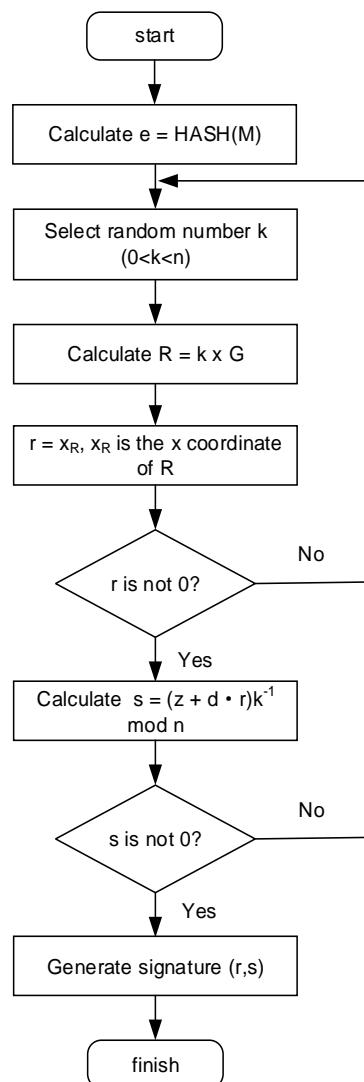
29.3.3. ECC algorithm

Suppose the message is M, d is the private key, G is the base point of the chosen elliptic curve, Q is a point of the chosen elliptic curve, with a prime order n. The hash function is HASH(), z is the Ln leftmost bits of HASH(M), where Ln is the bit length of the order n. The ECDSA sign and verification are detailed as follows:

ECDSA sign

The signature result of ECDSA consists of r and s. The process to generate ECDSA signature is shown in [Figure 29-3. Flow chart of ECDSA sign](#).

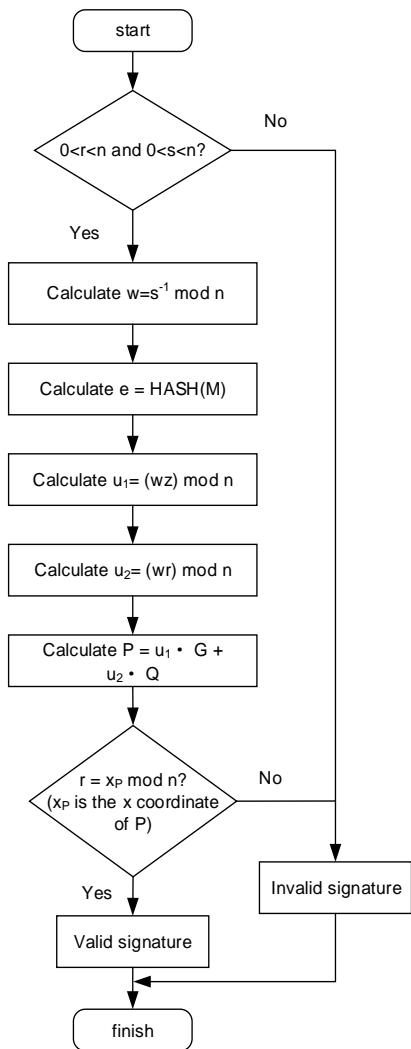
Figure 29-3. Flow chart of ECDSA sign



ECDSA verification

Before verifying the signature, be sure to get the signer's public key, message, and signature. The process to generate ECDSA signature is shown in [Figure 29-4. Flow chart of ECDSA verification](#).

Figure 29-4. Flow chart of ECDSA verification



Note: The HASH in the above diagram is the agreed cryptographic hash function.

29.3.4. Integer arithmetic operations

The integer arithmetic operation can be selected by configuring the MODSEL[5:0] in PKCAU_CTL register. The operation modes to be selected is shown in [Table 29-2. Integer arithmetic operations](#).

Table 29-2. Integer arithmetic operations

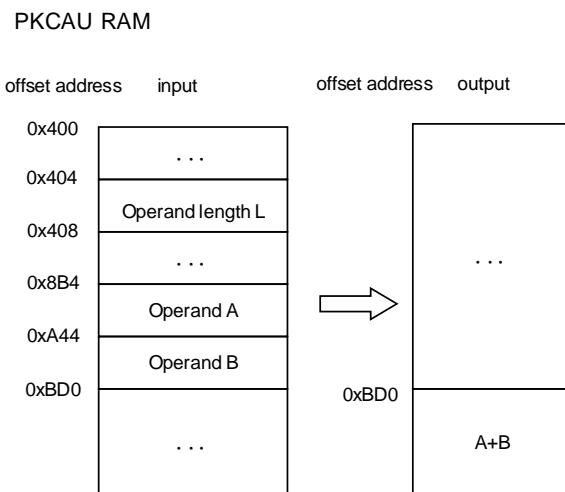
MODSEL[5:0]	Operation modes
000000	calculate Montgomery parameter and then modular exponentiation

MODSEL[5:0]	Operation modes
000001	calculate Montgomery parameter only
000010	modular exponentiation (the Montgomery parameter must be preloaded)
000111	RSA CRT exponentiation
001000	Modular inversion
001001	Arithmetic addition
001010	Arithmetic subtraction
001011	Arithmetic multiplication
001100	Arithmetic comparison
001101	Modular reduction
001110	Modular addition
001111	Modular subtraction
010000	Montgomery multiplication

Arithmetic addition

The arithmetic addition operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001001". The operation declaration is shown in [Figure 29-5. Arithmetic addition](#). The operation result is "result = A+B".

Figure 29-5. Arithmetic addition



$$0 \leq A < 2^L, 0 \leq B < 2^L, 0 \leq \text{result} < 2^{L+1}, 0 < L \leq 3136.$$

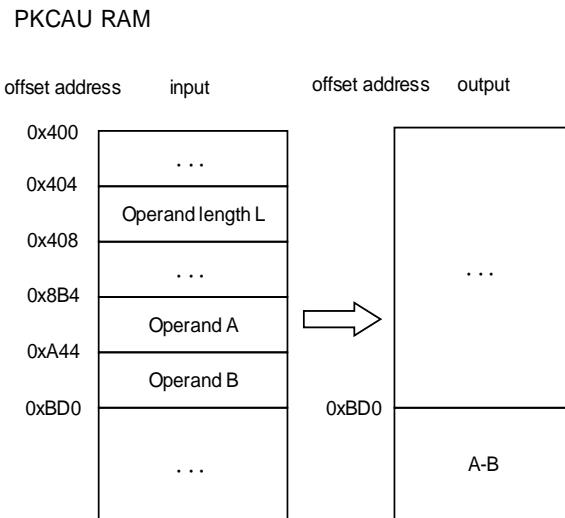
Arithmetic subtraction

The arithmetic subtraction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001010". The operation declaration is shown in [Figure 29-6. Arithmetic subtraction](#).

If $A \geq B$, the operation result is "result = A-B";

If $A < B$, the operation result is "result = A-B+2^{L+\lceil L/32 \rceil}"

Figure 29-6. Arithmetic subtraction

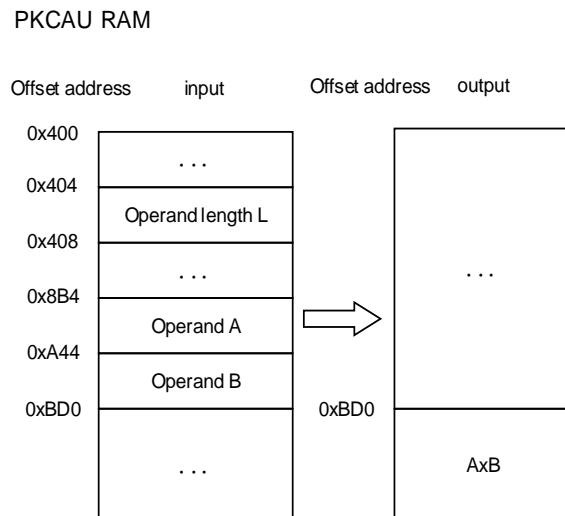


$$0 \leq A < 2^L, 0 \leq B < 2^L, 0 \leq \text{result} < 2^L, 0 < L \leq 3136.$$

Arithmetic multiplication

The arithmetic multiplication operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001011". The operation declaration is shown in [Figure 29-7. Arithmetic multiplication](#). The operation result is "result = A×B".

Figure 29-7. Arithmetic multiplication



$$0 \leq A < 2^L, 0 \leq B < 2^L, 0 \leq \text{result} < 2^{2L}, 0 < L \leq 3136.$$

Arithmetic comparison

The arithmetic comparison operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001100". The operation declaration is shown in [Figure 29-8. Arithmetic comparison](#).

comparison.

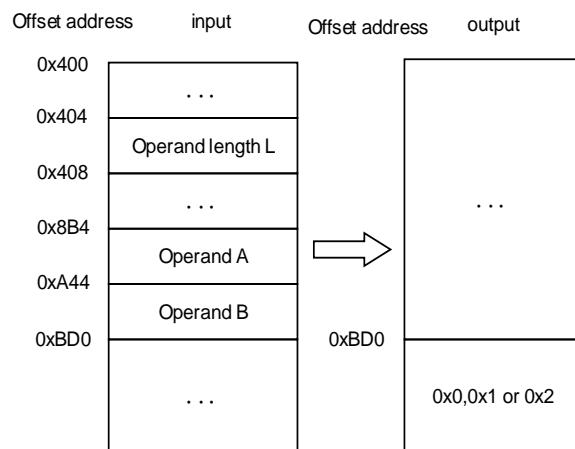
If A=B, the operation result is “result =0x0”;

If A>B, the operation result is “result =0x1”;

If A<B, the operation result is “result =0x2”.

Figure 29-8. Arithmetic comparison

PKCAU RAM



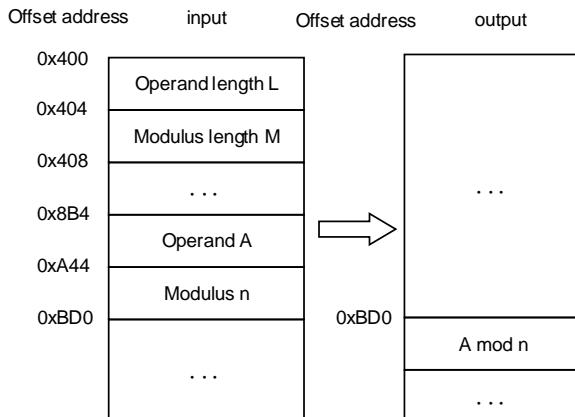
$0 \leq A < 2^L$, $0 \leq B < 2^L$, result =0x0, 0x1, 0x2, $0 < L \leq 3136$.

Modular reduction

The modular reduction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001101". The operation declaration is shown in [Figure 29-9. Modular reduction](#). The operation result is “result = A mod n”.

Figure 29-9. Modular reduction

PKCAU RAM

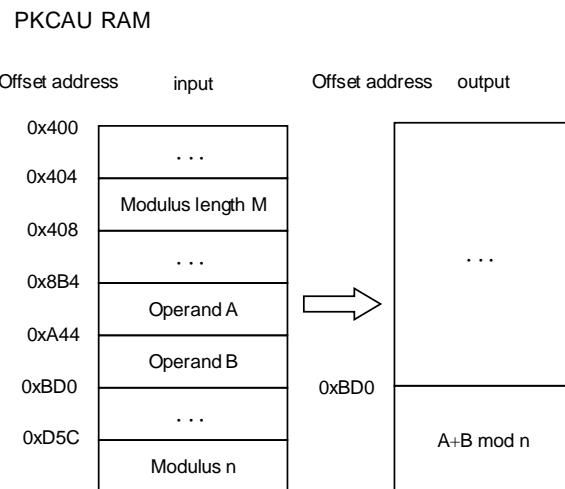


$0 < L \leq 3136$, $0 < M \leq 3136$, $0 \leq A < 2^L$, $0 < n < 2^M$, $0 \leq \text{result} < n$.

Modular addition

The modular addition operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001110". The operation declaration is shown in [Figure 29-10. Modular addition](#). The operation result is “result = A+B mod n”.

Figure 29-10. Modular addition



$$0 \leq A < n, 0 \leq B < n, 0 \leq \text{result} < n, 0 < n < 2^M, 0 < M \leq 3136.$$

Modular subtraction

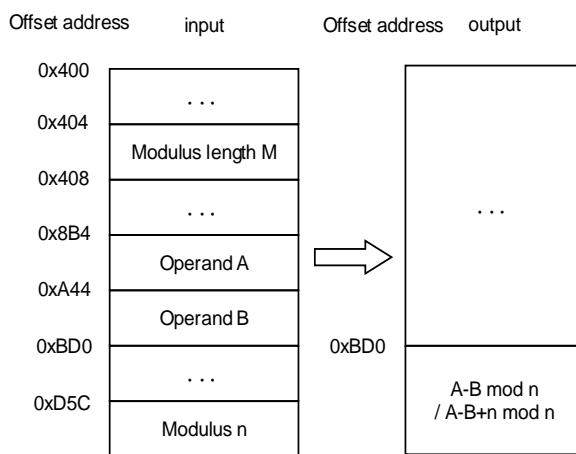
The modular subtraction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001111". The operation declaration is shown in [Figure 29-11. Modular subtraction](#).

If $A \geq B$, the operation result is “result = A-B mod n”;

If $A < B$, the operation result is “result = A-B+n mod n”.

Figure 29-11. Modular subtraction

PKCAU RAM



$$0 \leq A < n, 0 \leq B < n, 0 \leq \text{result} < n, 0 < n < 2^M, 0 < M \leq 3136$$

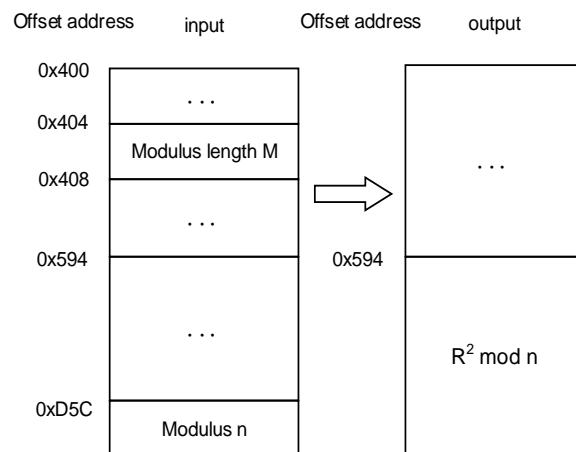
Montgomery parameter calculation

PKCAU needs to use the Montgomery parameter ($R^2 \bmod n$) to convert the operands to Montgomery residue system representation.

The Montgomery parameter calculation operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000001". The operation declaration is shown in [Figure 29-12. Montgomery parameter calculation](#).

Figure 29-12. Montgomery parameter calculation

PKCAU RAM



$$0 < M \leq 3136, 1 < n < 2^M (n \text{ must be odd integer}).$$

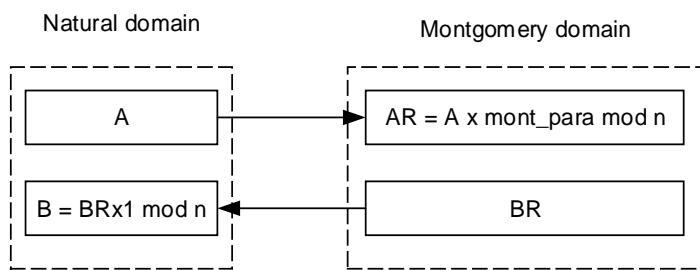
Montgomery multiplication

Suppose A, B and C are in natural domain. “x” function is Montgomery multiplication operation. The two main uses of this operation are as follows:

1. Mutual mapping between Montgomery domain and natural domain.

As is shown in [**Figure 29-13. Mutual mapping between Montgomery domain and natural domain**](#), if A is an integer in natural domain, the Montgomery parameter $\text{mont_para} = R^2 \bmod n$, the result $AR = A \times \text{mont_para} \bmod n$ is A in Montgomery domain. In turn, if BR is an integer in Montgomery domain, the calculation result $B = BR \times 1 \bmod n$ is in natural domain.

Figure 29-13. Mutual mapping between Montgomery domain and natural domain



2. Perform a modular multiplication operation $A \times B \bmod n$.

- (1) Calculate Montgomery parameter $\text{mont_para} = R^2 \bmod n$.
- (2) Calculate $AR = A \times \text{mont_para} \bmod n$, the output is in Montgomery domain.
- (3) Calculate $AB = AR \times B \bmod n$, the output is in natural domain.

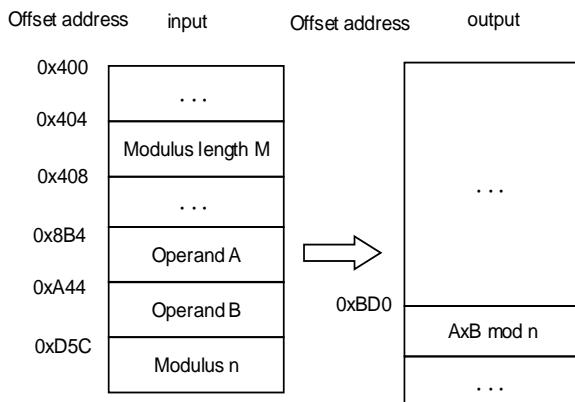
Multiple modular multiplication $A \times B \times C \bmod n$

- (1) Calculate Montgomery parameter $\text{mont_para} = R^2 \bmod n$.
- (2) Calculate $AR = A \times \text{mont_para} \bmod n$, the output is in Montgomery domain.
- (3) Calculate $BR = B \times \text{mont_para} \bmod n$, the output is in Montgomery domain.
- (4) Calculate $ABR = AR \times BR \bmod n$, the output is in Montgomery domain.
- (5) Calculate $CR = C \times \text{mont_para} \bmod n$, the output is in Montgomery domain.
- (6) Calculate $ABCR = ABR \times CR \bmod n$, the output is in Montgomery domain.
- (7) Calculate $ABC = ABCR \times 1 \bmod n$, the output is in natural domain.

The Montgomery multiplication operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "010000". The operation declaration is shown in [**Figure 29-14. Montgomery multiplication**](#).

Figure 29-14. Montgomery multiplication

PKCAU RAM



$0 \leq A < n$, $0 \leq B < n$, $0 < n < 2^M$, $0 < M \leq 3136$ (n must be odd integer).

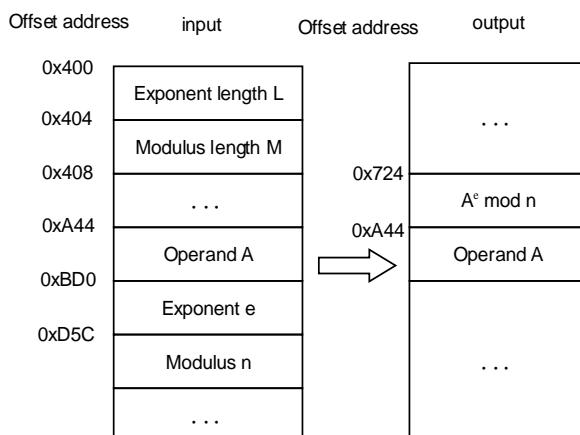
Modular exponentiation

Normal mode

The Modular exponentiation of normal mode operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000000". The operation declaration is shown in [Figure 29-15. Modular exponentiation of normal mode](#). The operation result is "result = $A^e \bmod n$ ".

Figure 29-15. Modular exponentiation of normal mode

PKCAU RAM



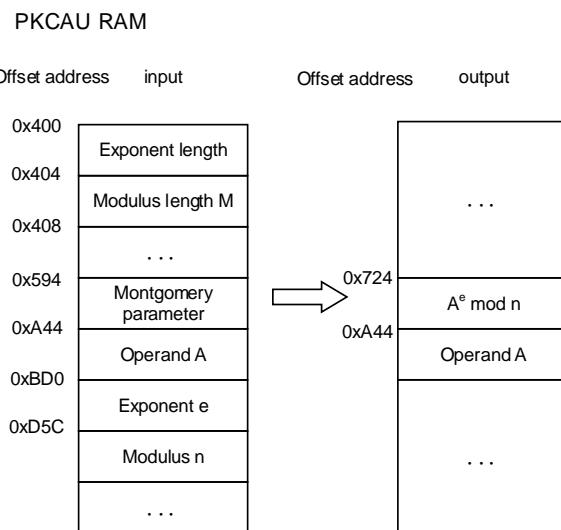
$0 < L \leq 3136$, $0 < M \leq 3136$, $0 \leq A < n$, $0 \leq e < 2^L$, $0 \leq \text{result} < n$, $1 < n < 2^M$ (n must be odd integer).

Fast mode

The Modular exponentiation of fast mode operation is selected by configuring MODSEL[5:0]

in PKCAU_CTL register as "000010". The operation declaration is shown in [Figure 29-16](#). **Modular exponentiation of fast mode**. The operation result is “result = $A^e \bmod n$ ”.

Figure 29-16. Modular exponentiation of fast mode

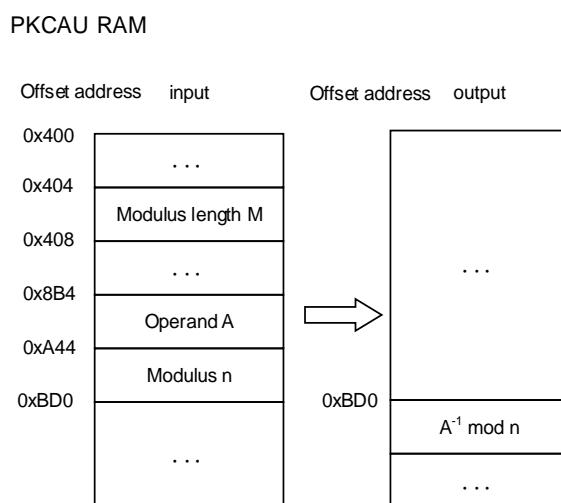


$0 \leq A < n$, $0 \leq e < n$, $0 \leq \text{result} < n$, $0 < n < 2^M$, $0 < M \leq 3136$, $0 < \text{Montgomery parameter } (R^2 \bmod n) < n$.

Modular inversion

The Modular exponentiation of fast mode operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001000". The operation declaration is shown in [Figure 29-17](#). **Modular inversion**. The operation result is “result = $A^{-1} \bmod n$ ”.

Figure 29-17. Modular inversion



$0 < A < n$, $0 < \text{result} < n$, $0 < n < 2^M$, $0 < M \leq 3136$.

Note:

1. If the modulus n is prime, for all values of A that satisfy the condition “ $1 \leq A < n$ ”, the modular inversion output is valid.

2. If the modulus n is not prime, only when the greatest common divisor of A and n is 1, the modular inversion output is valid.

RSA CRT exponentiation

The RSA CRT exponentiation operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000111".

p and q are part of the private key, and are primes

$$d_p = d \bmod (p-1)$$

$$d_q = d \bmod (q-1)$$

$$q_{inv} = q^{-1} \bmod p$$

These parameters above allow the recipient to compute the exponentiation $m = A^d \pmod{pq}$ more efficiently as follows:

$$m = A^d \pmod{pq}$$

$$m_1 = A^{dp} \pmod{p}$$

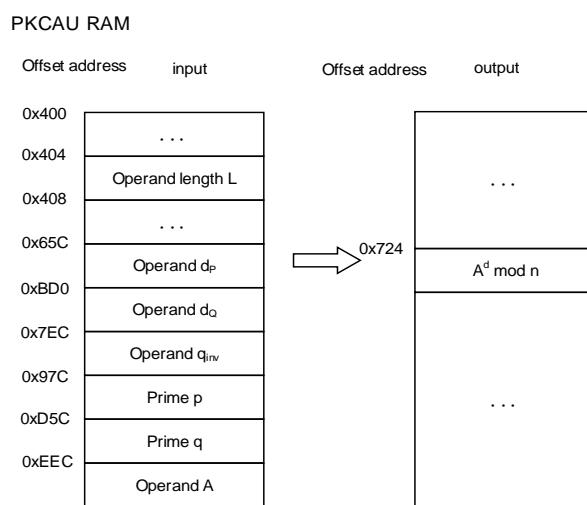
$$m_2 = A^{dq} \pmod{p}$$

$$h = q_{inv} (m_1 - m_2) \bmod p, m_1 > m_2$$

$$m = m_2 + hq$$

The operation declaration is shown in [Figure 29-18. RSA CRT exponentiation](#). The operation result is “result = $A^d \pmod{pq}$ ”.

Figure 29-18. RSA CRT exponentiation



The range of parameters used by RSA CRT exponentiation operation is shown in [Table 29-3. Range of parameters used by RSA CRT exponentiation operation](#).

Table 29-3. Range of parameters used by RSA CRT exponentiation operation

Parameters		Range
Input	Operand d_P	$0 \leq d_P < 2^{L/2}$
	Operand d_Q	$0 \leq d_Q < 2^{L/2}$
	Operand q_{inv}	$0 < q_{inv} < 2^{L/2}$
	Prime p	$0 < p < 2^{L/2}$
	Prime q	$0 < q < 2^{L/2}$
	Operand A	$0 \leq A < 2^L$
Output	$result = A^d \bmod pq$	$0 \leq result < pq$

29.3.5. Elliptic curve operations in Fp domain

The Elliptic curve operation mode in Fp domain can be selected by configuring the MODSEL[5:0] in PKCAU_CTL register. The operation modes to be selected is shown in [Table 29-4. Elliptic curve operations in Fp domain](#).

Table 29-4. Elliptic curve operations in Fp domain

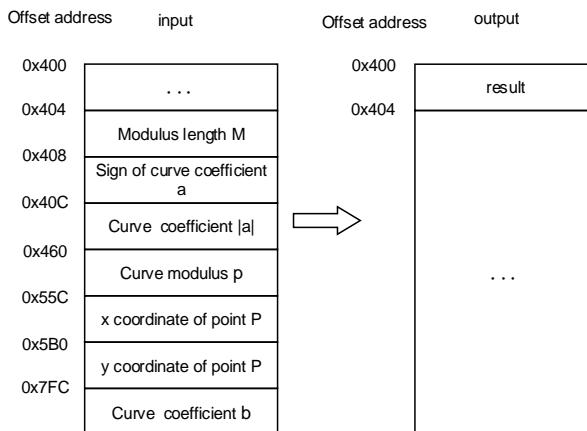
MODSEL[5:0]	Operation modes
100000	Montgomery parameter computation then ECC scalar multiplication
100010	ECC scalar multiplication only (Montgomery parameter must be loaded first)
100100	ECDSA sign
100110	ECDSA verification
101000	Point on elliptic curve Fp check

Point on elliptic curve Fp check

The operation is used to check whether $P(x,y)$ is on the $y^2 = x^3 + ax + b \bmod p$ in prime domain, where A and B are curve coefficients. The point on elliptic curve check operation in Fp domain is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "101000". The operation declaration is shown in [Figure 29-19. Point on elliptic curve Fp check](#). If the operation result is 0, it indicates that point P is on the elliptic curve. Or else, it indicates that point P is not on the elliptic curve.

Figure 29-19. Point on elliptic curve Fp check

PKCAU RAM



The range of parameters used by point on elliptic curve Fp check operation is shown in [Table 29-5. Range of parameters used by point on elliptic curve Fp check](#).

Table 29-5. Range of parameters used by point on elliptic curve Fp check

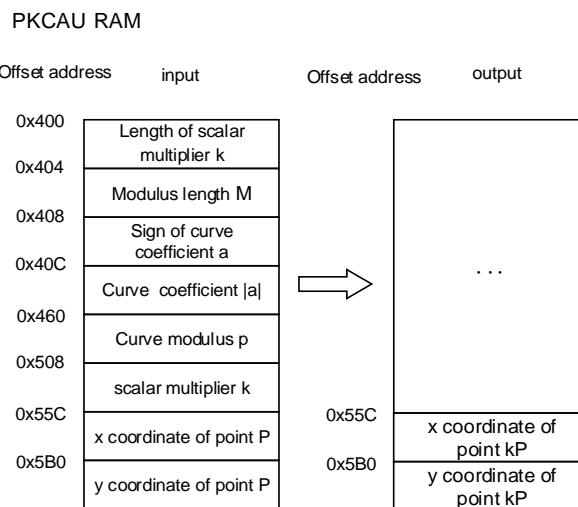
Parameters		Range
Input	Modulus length M	$0 < M \leq 640$
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient a	Absolute value $ a < p$
	Curve coefficient b	Absolute value $ b < p$
	Curve modulus p	Odd prime $0 < p \leq 2^M$
	x coordinate of point P	$x < p$
	y coordinate of point P	$y < p$

ECC scalar multiplication

ECC scalar multiplication operation is $ak \times P(x_P, y_P)$, where P is a point on the elliptic curve in the prime domain Fp. The operation result is also on the elliptic curve, or a point at infinity.

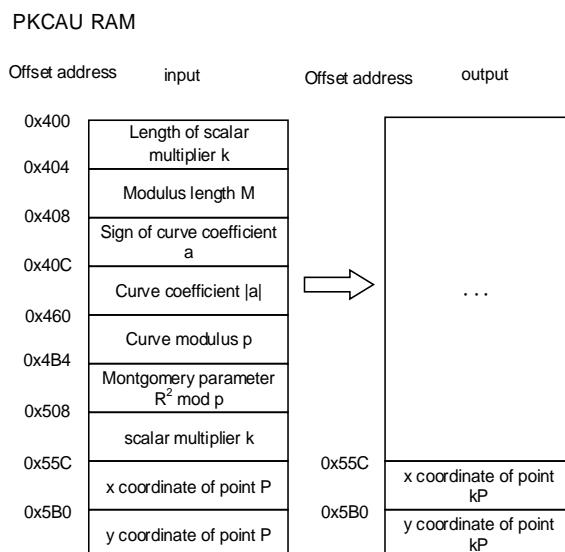
Normal mode

The ECC scalar multiplication operation in normal mode is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100000". The operation declaration is shown in [Figure 29-20. ECC scalar multiplication of normal mode](#).

Figure 29-20. ECC scalar multiplication of normal mode


Fast mode

The ECC scalar multiplication operation in fast mode is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100010". The operation declaration is shown in [Figure 29-21. ECC scalar multiplication of fast mode](#).

Figure 29-21. ECC scalar multiplication of fast mode


The range of parameters used by point on elliptic curve Fp check operation is shown in [Table 29-6. Range of parameters used by ECC scalar multiplication](#).

Table 29-6. Range of parameters used by ECC scalar multiplication

Parameters		Range
input	Length of scalar multiplier k (LEN)	$0 < \text{LEN} \leq 640$
	Modulus length M	$0 < M \leq 640$
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient a	Absolute value $ a < p$
	Curve modulus p	Odd prime $0 < p \leq 2^M$
	scalar multiplier k	$0 \leq k < 2^{\text{LEN}}$ ($k < n$, and n is the curve prime order)
	x coordinate of point P	$x_P < p$
output	y coordinate of point P	$y_P < p$
	x coordinate of point kP	$x < p$
	y coordinate of point kP	$y < p$

If $k=0$, the output is a point at infinity. When k is a multiple of curve prime order n , the output will also be a point at infinity. In this module, output is $(0, 0)$ if the result is a point at infinity.

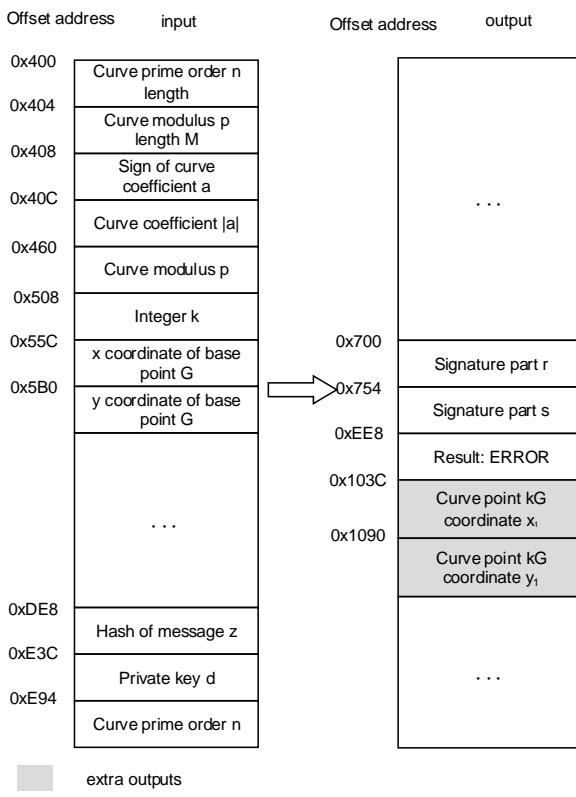
If $k < 0$, the absolute value of k replaces k as the scalar multiplier for ECC scalar multiplication. After the computation is completed, $-P = (x, -y)$ can be used to compute the final result of y .

ECDSA sign

The ECDSA sign operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100100". The operation declaration is shown in [Figure 29-22. ECDSA sign](#).

Figure 29-22. ECDSA sign

PKCAU RAM



The range of parameters used ECDSA sign operation is shown in [Table 29-7. Range of parameters used by ECDSA sign](#).

Table 29-7. Range of parameters used by ECDSA sign

Parameters	Range
input	Curve prime order n length (LEN)
	$0 < \text{LEN} \leq 640$
	Curve modulus p length (M)
	$0 < M \leq 640$
	Sign of curve coefficient a
	0x0: positive 0x1: negative
	Curve coefficient a
	Absolute value $ a < p$
	Curve modulus p
	Odd prime $0 < p < 2^M$
Integer k	$0 \leq k < 2^{\text{LEN}}$
x coordinate of base point G	$x < p$
y coordinate of base point G	$y < p$
Hash of message z	$z < 2^{\text{LEN}}$
Private key d	positive integer $d < n$
Curve prime order n	prime $n < 2^{\text{LEN}}$

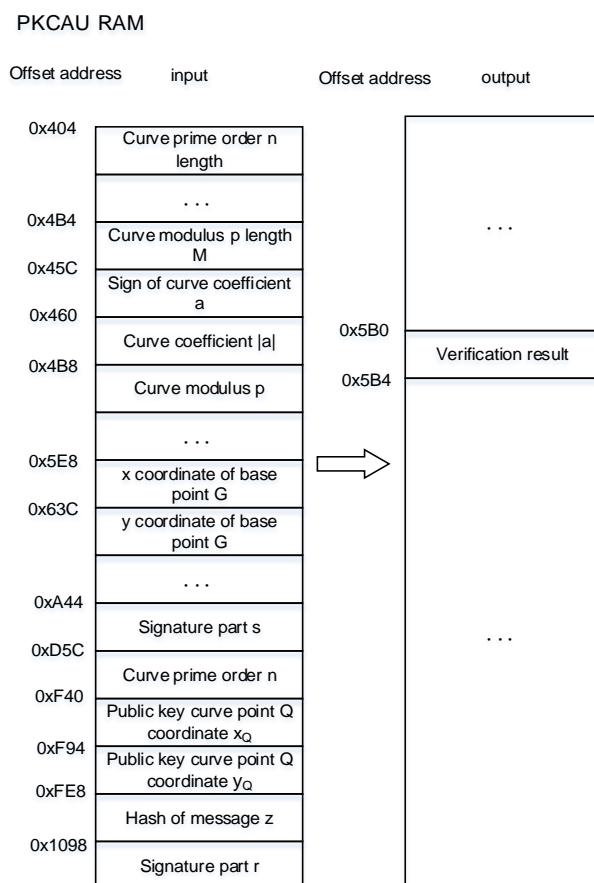
Parameters		Range
output	Signature part r	0<r<n
	Signature part s	0<s<n
	Signature result: ERROR	0x0: no error 0x1: Signature part r is 0 0x2: Signature part s is 0
	Curve point kG coordinate x ₁	0≤x ₁ <n
	Curve point kG coordinate y ₁	0≤y ₁ <n

If the error output is not 0, the content in PKCAU RAM will be cleared by hardware to avoid leaking private key related information.

ECDSA verification

The ECDSA verification operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100110". The operation declaration is shown in [Figure 29-23. ECDSA verification](#).

Figure 29-23. ECDSA verification



The range of parameters used ECDSA verification operation is shown in [Table 29-8. Range](#)

[of parameters used by ECDSA verification.](#)
Table 29-8. Range of parameters used by ECDSA verification

Parameters	Range
input	Curve prime order n length (LEN)
	0<LEN≤640
	Curve modulus p length (M)
	0<M≤640
	Sign of curve coefficient a
	0x0: positive 0x1: negative
	Curve coefficient a
	Absolute value a <p
	Curve modulus p
	Odd prime 0<p< 2^M
x coordinate of base point G	x<p
	y coordinate of base point G
Public key curve point Q coordinate x _Q	y<p
	x _Q <p
Public key curve point Q coordinate y _Q	y _Q <p
	Signature part r
Signature part s	0<r<n
	0<s<n
Hash of message z	z< 2^{LEN}
Curve prime order n	prime n< 2^{LEN}
output	verification result 0x0: verification valid Not 0x0: verification invalid

29.3.6. PKCAU operation process

The PKCAU can be enabled by setting the PKCAUEN bit in PKCAU_CTL register. When the PKCAU is performing a calculation, the PKCAUEN should not be cleared, or else the ongoing operation is terminated and the content in PKCAU RAM will not be guaranteed.

When PKCAUEN is 0, the application can still access PKCAU RAM through the AHB interface.

Operation process in normal mode

The flows below applies to all operations listed in MODSEL[5:0] bits in the PKCAU_CTL register.

1. After system reset, the PKCAU RAM is to be erased from the beginning to the end. During this period, the BUSY bit in PKCAU_STAT register is set. All operation to PKCAU RAM should not carry out until the BUSY bit is 0;

2. Load the initial data into PKCAU RAM at offset address 0x400;
3. Specify the operation to be performed in MODSEL[5:0] bits in PKCAU_CTL register, then set START bit in PKCAU_CTL register;
4. Wait for the ENDF bit set in PKCAU_STAT register;
5. Read calculation result from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register.

Operation process in fast mode

The fast mode computes the Montgomery parameter only once when calculating many operations with the same modulus. When the operation is performed, the pre-calculated Montgomery parameter is loaded to perform the calculation.

The flow of operation in fast mode is as follows:

1. Load the initial data into PKCAU RAM at offset address 0x400;
2. Select the Montgomery parameter calculation mode by configuring the MODSEL[5:0] as “000001”, then set START bit in PKCAU_CTL register;
3. Wait for the ENDF bit set in PKCAU_STAT register;
4. Read the Montgomery parameter from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register;
5. Load the initial data and Montgomery parameter into PKCAU RAM;
6. Specify the operation to be performed in MODSEL[5:0] bits in PKCAU_CTL register, then set START bit in PKCAU_CTL register;
7. Wait for the ENDF bit set in PKCAU_STAT register;
8. Read calculation result from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register.

29.3.7. Processing times

The following tables summarize the PKCAU computation times, expressed in clock cycles.

Table 29-9. Modular exponentiation computation times

Exponent length (in bits)	Mode	Operand length (in bits)		
		1024	2048	3072
1024	Normal	6780000	-	-
	Fast	6701000	-	-
	CRT	1853000	-	-
2048	Normal	-	52196000	-
	Fast	-	51910000	-

Exponent length (in bits)	Mode	Operand length (in bits)		
		1024	2048	3072
	CRT	-	13651000	-
3072	Normal	-	-	182783000
	Fast	-	-	181953000
	CRT	-	-	44905000

Table 29-10. ECC scalar multiplication computation times

Mode	Modulus length (in bits)					
	160	192	256	320	384	512
Normal	626000	951000	1997000	3617000	5762000	13134000
Fast	623000	946000	1990000	3607000	5749000	13111000

Table 29-11. ECDSA signature average computation times

Modulus length (in bits)					
160	192	256	320	384	512
634000	966000	2029000	3648000	5833000	13177000

Table 29-12. ECDSA verification average computation times

Modulus length (in bits)					
160	192	256	320	384	512
1261000	1901000	3997000	7225000	11477000	26287000

Table 29-13. Montgomery parameters average computation times

Modulus length (in bits)								
160	192	256	320	384	512	1024	2048	3072
3873	4658	7109	10330	14526	22301	79116	284359	626909

29.3.8. Status, errors and interrupts

There are several status and error flags in PKCAU, and interrupt may be asserted from these flags by setting some register bits.

- Access address error (ADDRERR)

When the accessed address exceeds the expected range of PKCAU RAM, the address error flag ADDRERR bit in PKCAU_STAT register will be set. If the ADDRERRIE bit is set in PKCAU_CTL register, an interrupt will be generated. The ADDRERR bit can be cleared by setting the ADDRERRC bit in PKCAU_STATC register.

- RAM error (RAMERR)

The PKCAU core is using the PKCAU RAM while an AHB access to the PKCAU RAM occurs, the RAM error flag RAMERR bit in PKCAU_STAT register will be set. If it is an AHB read, it returns 0, an AHB write will be ignored. If the RAMERRIE bit is set in PKCAU_CTL register, an interrupt will be generated. The RAMERR bit can be cleared by setting the RAMERRC bit in PKCAU_STATC register.

- End of operation flag (ENDF)

When the operation specified in MODSEL[5:0] bits in the PKCAU_CTL register is completed, the ENDF bit will be set. If the ENDIE bit in PKCAU_CTL register is set, an interrupt will be generated. The ENDF bit can be cleared by setting the ENDFC bit in PKCAU_STATC register. The ENDF bit can also be cleared automatically if another operation is carried out by set the START bit.

The PKCAU interrupt events and flags are listed in [Table 29-14. PKCAU interrupt requests](#)

Table 29-14. PKCAU interrupt requests

Interrupt event	Event flag	Flag clear	Enable control bit
Access address error	ADDRERR	ADDRERRC	ADDRERRIE
RAM error	RAMERR	RAMERRC	RAMERRIE
Operation end flag	ENDF	ENDFC	ENDIE

29.4. Register definition

PKCAU Secure access base address: 0x5C06 1000

PKCAU Non-secure access base address: 0x4C06 1000

29.4.1. Control register (PKCAU_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								ADDRER	RAMERR	Reserved		ENDIE	Reserved		
								RIE	IE						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		MODSEL[5:0]								Reserved				START	PKCAUE N

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	ADDRERRIE	Address error interrupt enable 0: Address error interrupt enable 1: Address error interrupt disable
19	RAMERRIE	RAM error interrupt enable 0: RAM error interrupt enable 1: RAM error interrupt disable
18	Reserved	Must be kept at reset value.
17	ENDIE	End of operation interrupt enable 0: End of operation interrupt enable 1: End of operation interrupt disable
16:14	Reserved	Must be kept at reset value.
13:8	MODSEL[5:0]	PKCAU operation mode selection 000000: Montgomery parameter computation then modular exponentiation 000001: Montgomery parameter computation only 000010: Modular exponentiation only (Montgomery parameter must be loaded first) 000111: RSA CRT exponentiation 001000: Modular inversion

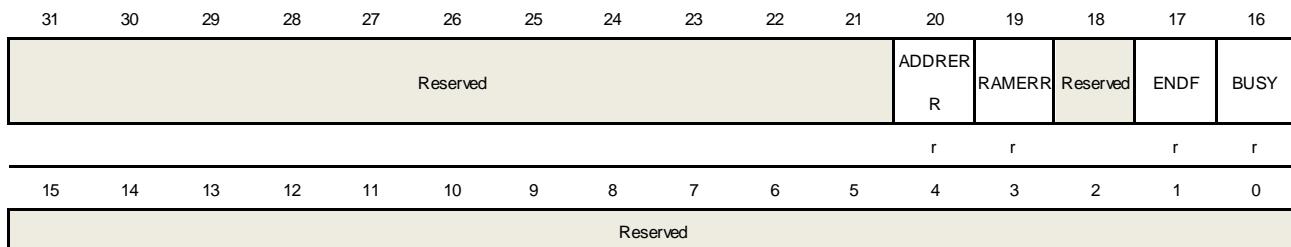
		001001: Arithmetic addition
		001010: Arithmetic subtraction
		001011: Arithmetic multiplication
		001100: Arithmetic comparison
		001101: Modular reduction
		001110: Modular addition
		001111: Modular subtraction
		010000: Montgomery multiplication
		100000: Montgomery parameter computation then ECC scalar multiplication
		100010: ECC scalar multiplication only (Montgomery parameter must be loaded first)
		100100: ECDSA sign
		100110: ECDSA verification
		101000: Point on elliptic curve Fp check
		Other values are reserved.
7:2	Reserved	Must be kept at reset value.
1	START	PKCAU starts operation This bit is set by software to start the PKCAU operation which is specified in MODSEL[5:0] in PKCAU_CTL register. When the BUSY bit in PKCAU_STAT register is 1, writing 1 to this bit will be ignored.
0	PKCAUEN	PKCAU enable 0: PKCAU disable 1: PKCAU enable

29.4.2. Status register (PKCAU_STAT)

Address offset: 0x04

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.

20	ADDRERR	Address error. 0: No address error. 1: The accessed address exceeds the expected range of PKCAU RAM, an address error occurs.
19	RAMERR	PKCAU RAM error 0: No PKCAU RAM error. 1: When the PKCAU core is using the RAM, AHB accesses the PKCAU RAM, a PKCAU RAM error occurs.
18	Reserved	Must be kept at reset value.
17	ENDF	End of PKCAU operation When the operation executed completely, this bit is set by hardware.
16	BUSY	Busy flag When the START bit in PKCAU_CTL register is set, this bit is set by hardware. When the PKCAU operation is completed, this bit is cleared by hardware.
15:0	Reserved	Must be kept at reset value.

29.4.3. Status clear register (PKCAU_STATC)

Address offset: 0x08

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
											ADDRERR	RAMERR	Reserved	ENDFC	Reserved
											RC	C			
											w	w		w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	ADDRERRRC	Address error flag clear. Software can clear the ADDRERR bit in PKCAU_STAT by writing 1 to this bit.
19	RAMERRRC	PKCAU RAM error flag clear. Software can clear the RAMERR bit in PKCAU_STAT by writing 1 to this bit.
18	Reserved	Must be kept at reset value.
17	ENDFC	End of PKCAU operation flag clear.

Software can clear the ENDF bit in PKCAU_STAT by writing 1 to this bit.

16:0	Reserved	Must be kept at reset value.
------	----------	------------------------------

30. High-Performance Digital Filter (HPDF)

30.1. Overview

A high performance digital filter module (HPDF) for external sigma delta ($\Sigma-\Delta$) modulator is integrated in GD32W51x. HPDF supports SPI interface and Manchester-coded single-wire interface. The external sigma delta modulator can be connected with MCU by the serial interface, and the serial data stream output by sigma delta modulator can be filtered. In addition, HPDF also supports the parallel data stream input function to filter the data in the internal memory of the MCU.

30.2. Characteristics

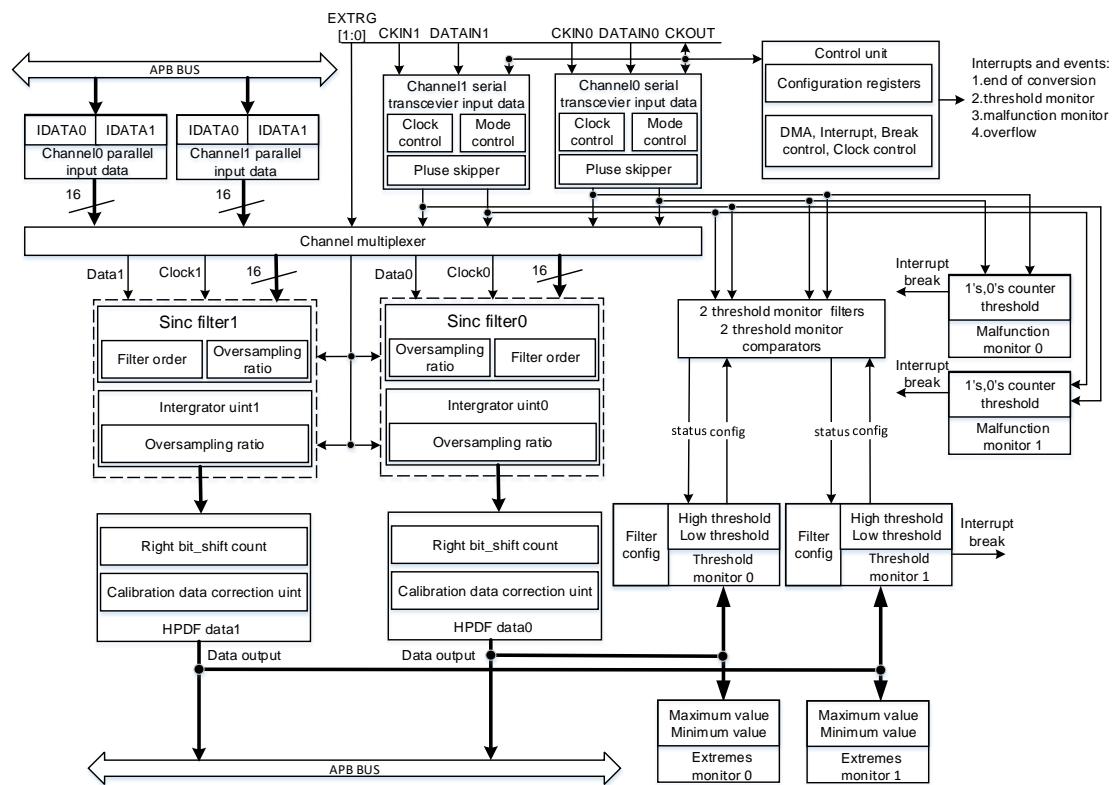
- Two multiplex digital serial input channels
 - configurable SPI and Manchester interfaces;
- Two internal digital parallel input channels
 - input with up to 16-bit resolution;
- Configurable Sinc filter and integrator
 - the order and oversampling rate (decimation rate) of Sinc filter can be configured;
 - sampling rate of configurable integrator;
- Threshold monitor function
 - independent Sinc filter, configurable order and oversampling rate (decimation rate);
 - configurable data input source: serial channel input data or HPDF output data;
- Malfunction monitor function
 - A counter with 8 bits is used to monitor the continuous 0 or 1 in the serial channel input data stream;
- Extreme monitor function
 - store minimum and maximum values of output data values of HPDF;
- Up to 24-bit output data resolution
- Clock signal can be provided to external sigma delta modulator
 - provide configurable clock signal by the CKOUT pin;
- Flexible conversion configuration function
 - the conversion channel is divided into regular group and inserted group;
 - support multiple conversion modes and startup modes;
- HPDF output data is in signed format

30.3. Function overview

30.3.1. HPDF Block Diagram

The structural block diagram of HPDF is shown in [Figure 30-1. HPDF block diagram](#).

Figure 30-1. HPDF block diagram



The HPDF interface communicates with the external $\Sigma\Delta$ modulator by the pins in [Table 30-1. HPDF pins definition](#).

Table 30-1. HPDF pins definition

PINs	Type	Description
EXTRG[1:0]	External trigger input	Input pin of external trigger signal source, the trigger signal sources are EXTI11 and EXTI15, which are used as the trigger signal of inserted group HPDF_ITRG[24] and HPDF_ITRG[25].
CKOUT	Clock out	The clock output signal of HPDF module, provides clock signal to external $\Sigma\Delta$ modulator.
CKINx	Clock input	External $\Sigma\Delta$ modulator provides clock signal to serial interface.
DATAINx	Data input	The external $\Sigma\Delta$ modulator transmits 1 bit data stream to the serial channel by this pin.

30.3.2. HPDF on-off control

When the HPDF module is started normally, the HPDF module can be enabled globally by setting HPDFEN to 1 in the HPDF_CH0CTL register. Then set the CHEN bit in HPDF_CHxCTL and the FLTEN bit in HPDF_FLTycCTL0 to 1 to enable the input channel and

channel digital filter respectively. In addition, as long as the input channel is enabled, the input channel will immediately start receiving serial data.

HPDF can enter stop mode by clearing FLTEN during operation. After entering stop mode, the ongoing conversion tasks of the HPDF module will immediately stop, and the configuration of the registers remains unchanged (except for the HPDF_FLTySTAT and HPDF_FLTyTMSTAT registers are reset).

In stop mode, the HPDF system clock will automatically stop. The HPDFEN bit must be cleared before the system clock is stopped to enter stop mode.

Low power mode

HPDF module optimizes the reduction of power consumption. In the normal working mode, the filter and integrator will automatically enter the idle state to achieve the purpose of reducing power consumption when there is no conversion task.

30.3.3. HPDF clock

The clock of HPDF includes the system clock and the serial clock. The system clock is used to drive the internal modules, and the serial clock used by the serial interface.

System clock

The system clock f_{HPDFCLK} of HPDF is used to drive channel transceiver, digital filter, integrator, threshold monitor, malfunction monitor, extremum detector and control module. The HPDF system clock source can be configured by the HPDFSEL bit in the ADDCTL register of the RCU chapter.

Serial input clock

The serial interface of HPDF can receive clock signal from external sigma delta modulator by CKINx pin, so as to receive the serial data stream from sigma delta modulator.

Using external input clock in serial interface is limited by clock frequency. If the standard SPI interface is used, the system clock $f_{\text{HPDFCLK}} \geq 4f_{\text{CKIN}}$. If the Manchester coding interface is used, the system clock $f_{\text{HPDFCLK}} \geq 6f_{\text{CKIN}}$ is required.

Serial output clock

HPDF supports the function of outputting serial clock, which can drive sigma delta modulator connected with it. The source of the serial output clock can be selected by CKOUTSEL bit in HPDF_CH0CTL register. When CKOUTSEL=0, the serial output clock source is the HPDF system clock. When CKOUTSEL=1, the serial output clock source is the audio clock. And the configuration of the audio clock can refer to the HPDFAUDIOSEL[1:0] bit field in the ADDCTL register of the RCU chapter.

After the serial output clock source is determined, the output clock frequency division can be controlled by configuring the CKOUTDIV [7:0] bit field in the HPDF_CH0CTL register. When CKOUTDIV[7:0] ≠ 0, the value of the serial output clock divider is CKOUTDIV[7:0]+1. When CKOUTDIV[7:0] = 0, the serial output clock is disabled and the pin of CKOUT remains low.

In addition, after clearing HPDFEN, the signal of serial output clock can also be stopped. When the serial output clock source is the system clock (CKOUTSEL = 0), if clear HPDFEN, the serial output clock stopped after 4 system clocks. When the serial output clock source is the audio clock (CKOUTSEL = 1), if clear HPDFEN, the serial output clock stopped after one system clock and three audio clocks.

The serial output clock source can only be modified when HPDFEN = 0. In order to avoid the burr signal on the pin of CKOUT, the software can only modify the value of the CKOUTSEL bit in the HPDF_CH0CTL register after the serial output clock stopped.

The frequency range of the serial output clock is 0-20MHz.

30.3.4. Multiplex serial data channel

HPDF has two multiplexing serial data channels, which support SPI code and Manchester code. The interface type supported can be selected for the current channel by configuring the SITYP[1:0] bit field in the HPDF_CHxCTL register.

SPI interface

Under the standard SPI interface, sigma delta modulator sends 1-bit data stream to the serial channel by the pin of DATAINx. The clock signal between HPDF and sigma delta modulator can be output by CKOUT pin or input by CKINx pin.

The data sampling point in SPI communication is determined by the SITYP[1:0] bit field and SPICKSS[1:0] bit field in HPDF_CHxCTL register. The data sampling points in SPI communication are shown in the table.

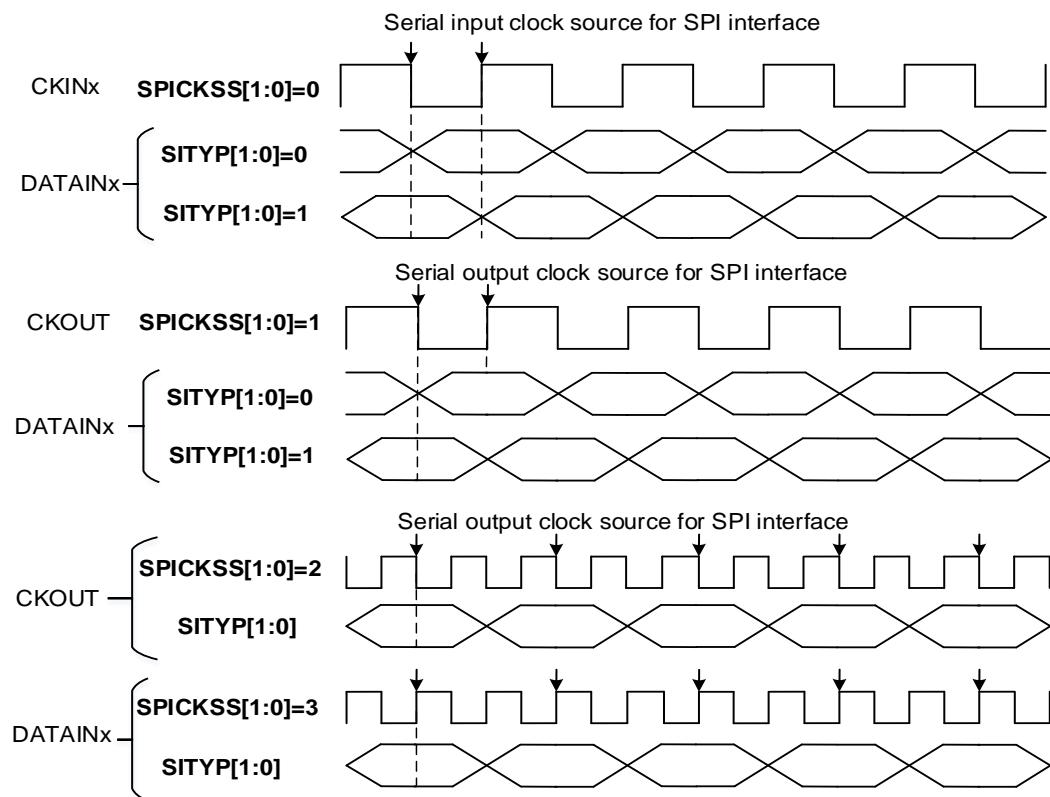
Table 30-2. SPI interface clock configuration

SPICKSS[1:0]	Clock source	SITYP[1:0]	Sampling point	Description
00	CKINx signal	00	rising edge	Data is sampled at the rising edge of the external serial input clock signal
		01	falling edge	Data is sampled at the falling edge of the external serial input clock signal
01	CKOUT signal	00	rising edge	The data is sampled at the rising edge of the internal serial output clock signal
		01	falling edge	The data is sampled at the falling edge of the internal serial output

SPICKSS[1:0]	Clock source	SITYP[1:0]	Sampling point	Description
				clock signal
10	CKOUT/2 signal (Generated at the rising edge of CKOUT)	xx	Rising edge of each second CKOUT signal	The external sigma delta modulator divides the CKOUT signal into 2 frequencies to generate the serial input communication clock. The data is sampled at the falling edge of every second CKOUT.
11	CKOUT/2 signal (Generated at the falling edge of CKOUT)	xx	Falling edge of each second CKOUT signal	The external sigma delta modulator divides the CKOUT signal into 2 frequencies to generate the serial input communication clock. The data is sampled at the rising edge of every second CKOUT.

According to [Table 30-2. SPI interface clock configuration](#), the sequence diagram of SPI data transmission is shown in the figure below.

Figure 30-2. The sequence diagram of SPI data transmission



Note: if SPI data interface is adopted, the frequency range of clock source is 0-20MHz and less than $f_{HDFCLK}/4$.

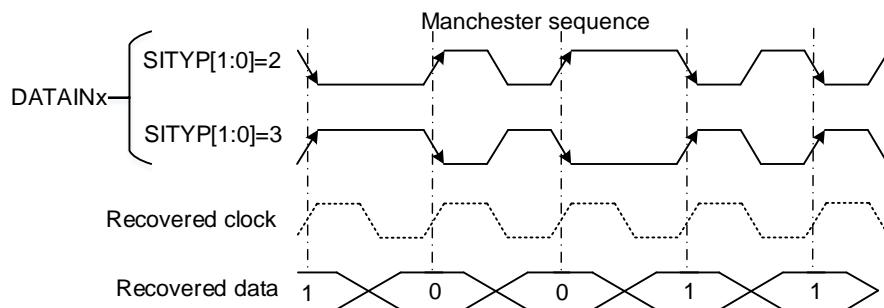
Manchester interface

When HPDF has two multiplexing serial data channels using Manchester encoding format, two encoding formats can be configured by SITYP[1:0] bit field in HPDF_CHxCTL:

1. When SITYP[1:0] = 2, Manchester code: rising edge = logic 0, falling edge = logic 1.
2. When SITYP[1:0] = 3, Manchester code: rising edge = logic 1, falling edge = logic 0.

When Manchester code is used, the data stream between the external sigma delta modulator and HPDF is only transmitted by the DATAINx pin. After the HPDF module Manchester decoding, the clock signal and data are recovered from the serial data stream. The recovered clock signal frequency must be between 0-10MHz and less than $f_{HPDFCLK}/6$. The timing chart of Manchester data transmission is shown in the figure below.

Figure 30-3. The sequence diagram of Manchester data transmission



In order to receive and decode Manchester data correctly, configure the CKOUTDIV[7:0] frequency divider according to the expected flow rate of Manchester data. The value of CKOUTDIV [7:0] is calculated with reference to the following format:

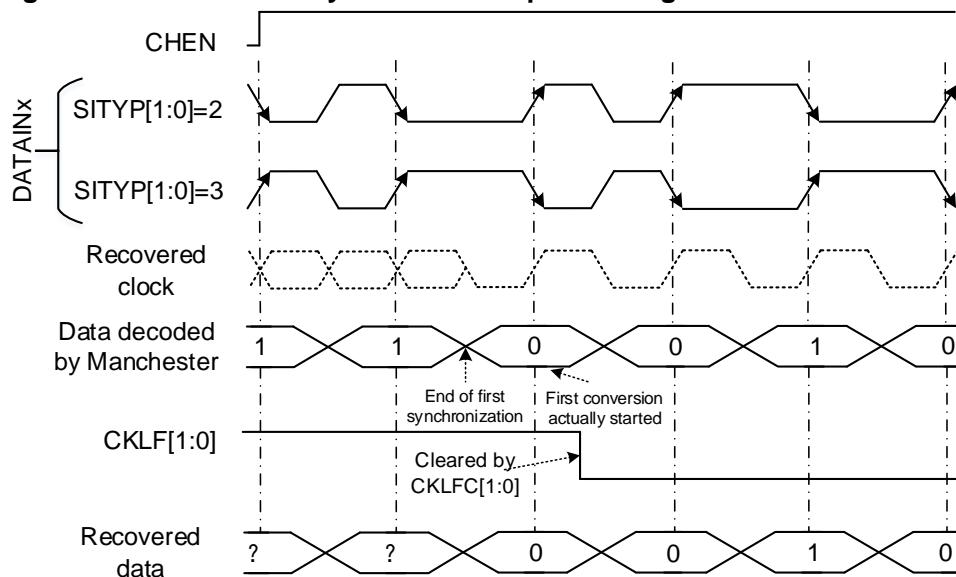
$$((CKOUTDIV+1) \times T_{HPDFCLK}) < T_{Manchester_clock} < (2 \times CKOUTDIV \times T_{HPDFCLK}) \quad (30-1)$$

Serial communication coding synchronization

After the serial channel is enabled, the data can only be received correctly after successful synchronization. The synchronization of SPI code occurs after the first detection of clock input signal by SPI data stream. If the channel uses Manchester coding, the first synchronization occurs when the channel receives data stream changes from 1-0 or 0-1.

Before the transceiver of the serial channel synchronizes, the clock loss flag bit of channel is set to 1. After successful synchronization, the clock loss flag bit can be cleared by CKLFC[1:0]. When the transceiver of the serial channel is not synchronized, the clock loss flag bit cannot be cleared by the CKLFC[1:0]. Therefore, it is possible to determine whether the serial channel is successfully synchronized by querying the CKLF[1:0] bit circularly. The following figure shows the timing chart of the first synchronization of Manchester code.

Figure 30-4. Manchester synchronous sequence diagram



Clock loss detection

Clock loss detection is to detect whether the channel serial input clock (CKINx signal) is lost, so as to ensure whether there is any error in the data of serial channel conversion (or threshold monitor and malfunction monitor). If a clock signal loss event occurs, the given data should be discarded. When using the clock loss detection function, you must configure the ckout signal source as the system clock.

The clock loss detection function can be enabled or disabled by the CKLEN bit in HPDF_CHxCTL register. When the enable clock loss detection function and the clock loss interrupt CKLIE occur, if a clock loss event occurs, the clock loss flag bit (CKLF) will be set to 1 and a clock loss interrupt will be generated. The corresponding interrupt flag bit can be cleared by setting CKLFC[1:0] bit field.

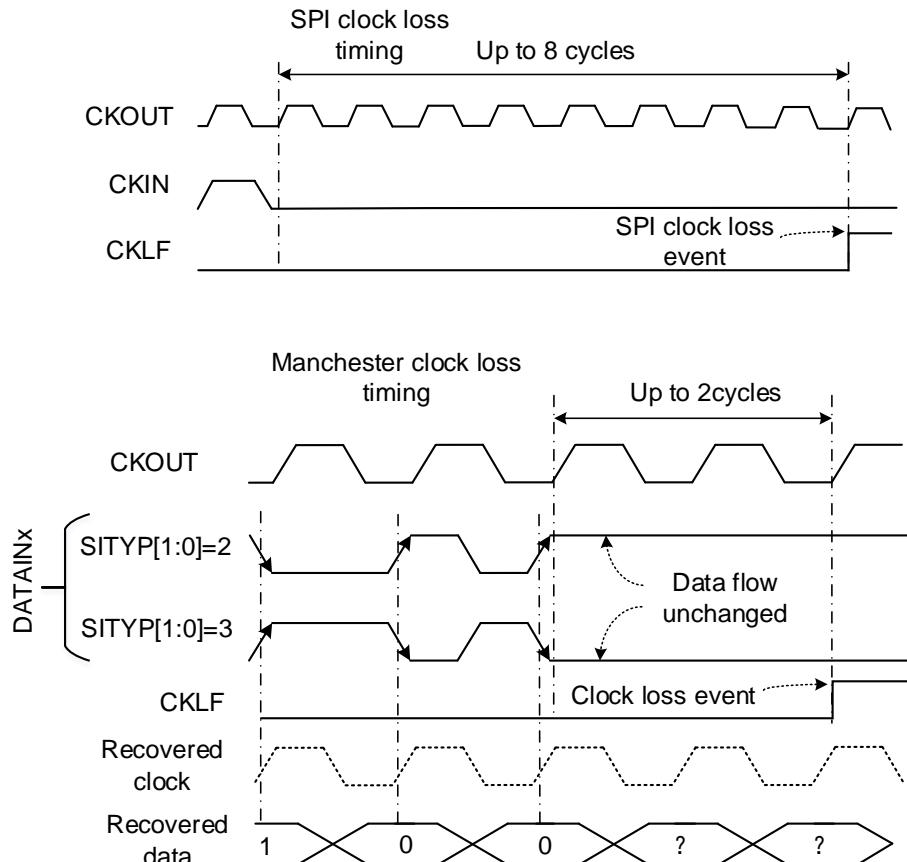
When the transceiver of the serial interface has not been synchronized, the clock loss flag bit is set to 1 and cannot be cleared by the corresponding CKLFC[1:0]. Therefore, the correct steps to use the clock loss function are as follows:

1. Enable the channel CHEN = 1.
2. Check the clock loss flag cyclically and write 1 to the CKLFC of the channel. When it is confirmed that the CKLF bit is cleared, the serial channel transceiver synchronization is successful.
3. Enable clock loss detection function CKLEN = 1. To detect possible clock loss, enable clock loss interrupt CKLIE = 1.

If the SPI interface is used in the serial channel, the external serial input clock (CKINx signal) is compared with the serial output clock (ckout signal) when the clock loss detection function is used. The external serial input clock signal must be inverted at least once every 8 CKOUT signal cycles, otherwise a clock loss event will occur.

If the serial channel uses the Manchester interface, the clock loss detection starts after the first successful synchronization of the Manchester code, and the external serial input data (DATAINx signal) is compared with the serial output clock (CKOUT signal). The serial input data must change every 2 ckout signal cycles, otherwise clock loss event will be generated. The timing of clock loss is shown in the figure below.

Figure 30-5. Clock loss detection timing diagram



Note: the maximum rate of Manchester encoded data stream must be less than the clock output CKOUT signal.

Channel pin redirection

Channel pin redirection means that the pins of serial channel 0 can be configured as the pins of channel 1, that is, channel 0 can read information from the DATAIN1 and CKIN1 pins. Pin redirection is used to sampling audio data of PDM microphone. The audio signal of PDM microphone includes data and clock signal. The data is divided into left/right channel data. The left channel data is sampled at the rising edge of clock signal, and the right channel data is sampled at the falling edge of clock signal.

When PDM microphone data stream is input into serial channel, its configuration process is as follows:

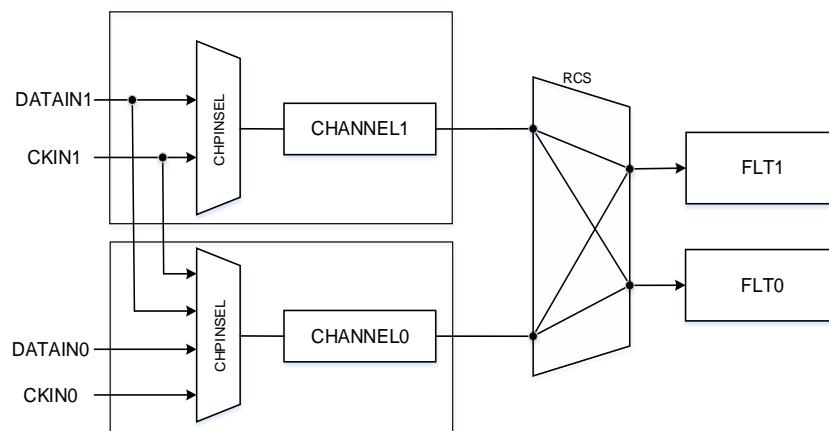
1. Select the HPDF serial channel 1 of PDM microphone data stream input.
2. Write 0 to CHPINSEL bit of channel 1 in HPDF_CHxCTL register, and input pin of channel

1 is own pin, DATAINx and CKINx. When SITYP[1:0] = 2b'00, the serial data stream is sampled at the rising edge of the clock signal, that is, the input of channel 1 is the left channel data.

3. Set the CHPINSEL to 1 in channel 0, and the DATAINx and CKINx pins will be used for channel 0. When SITYP[1:0] = 2b'01, the serial data stream is sampled at the falling edge of the clock signal, that is, the input of channel 0 is the right channel data.
4. Configure channel 0 and channel 1 with corresponding filters to filter the left and right channel data of PDM microphone.

The channel pin redirection diagram of HPDF module is shown in [Figure 30-6. Channel pins redirection.](#)

Figure 30-6. Channel pins redirection



Pulses skipper

When PCLK2 is used as the system clock source of HPDF, the pulse skipper function can be used. Pulse skipper refers to that the serial input data stream enters the filter after skipping a specified number of clock pulses, so as to discard a certain number of bit bits. This operation will cause the final output sample (and the next sample) from the filter to be calculated from the subsequent input data compared to the data stream that was not skipped.

The number of pulses to be skipped is determined by the PLSK[5:0] bit field in the HPDF_CHxPS register. Write the value to PLSK[5:0] bit field, and the specified channel will start to perform the pulse skipper function. Read the of PLSK[5:0], indicating the number of remaining pulse skipper not executed. For a single write operation of PLSK[5:0], the maximum number of pulse skipper executed is 63. More pulse skipper can be obtained by writing to PLSK[5:0] bit field several times.

Serial input interface configuration

The configuration steps of serial input interface of HPDF module are as follows:

1. Configure clock output prescaler: by configuring the CKOUTDIV[7:0] bit field in the

HPDF_CH0CTL register, the coefficient of prescaler is CKOUTDIV[7:0] + 1.

2. Configure the serial interface type and input clock phase: configure the serial interface type as SPI code or Manchester code, and determine the clock input sampling edge by the SITYP[1:0] bit field in HPDF_CHxCTL register.
3. Configure input clock source: select the clock source of serial interface as serial input clock or serial output clock by configuring SPICKSS[1:0] in HPDF_CHxCTL register.
4. Configure data offset correction and shift right bits: DTRS[4:0] defines the bits of the final data shift right in HPDF_CHxCFG register. After data shift, perform offset calibration defined by CALOFF [23:0] bit field.
5. Enable short circuit detection and clock loss detection function: enable short circuit detection and clock loss detection function by setting MMEN and CKLEN to 1.
6. Set the threshold monitor filter and malfunction monitor: the filter parameters of the threshold monitor, the malfunction signal allocation of the malfunction monitor and the counter threshold are all configured by the HPDF_CHxCFG1 register.

30.3.5. Parallel data input

HPDF module can select parallel data as the data input source of the channel. The CMSD[1:0] bit field in HPDF_CHxCTL is configured to determine whether the channel data input source is from serial data or parallel data. Each channel provides a 32-bit parallel data input register (HPDF_CHxPDI), which can write two 16 bit parallel data by CPU/DMA. The register has two 16-bit data in signed format.

CPU / DMA write parallel data

There are two ways to write parallel data: CPU direct write and DMA write. When using DMA to write parallel data, DMA should be configured as memory to memory mode, and its target address is the address of HPDF_CHxPDI.

Note: DMA writing parallel data is different from DMA reading final conversion data from HPDF module. The latter needs to be configured in peripheral to memory mode.

Parallel data packed mode

The data stored in HPDF_CHxPDI register will be processed by channel filter. There are three modes of parallel data stored in the HPDF_CHxPDI register. In different data packed modes, the number of filter samples allowed to load depends on the value of DPM [1:0] bit field in the HPDF_CHxCTL register. The different data encapsulation modes are as follows:

1. Standard mode (DPM[1:0] = 2'b00):

In this mode, the upper 16 bits in the HPDF_CHxPDI register are write protected, and the 16-bit data written by CPU/DMA is stored in the low 16 bit DATAIN0[15:0] bit field. CPU/DMA is configured as a 16-bit access mode. When writing 16-bit data once, the

channel filter must perform an input sampling to clear the HPDF_CHxPDI register.

2. Interleaving mode (DPM[1:0] = 2'b01):

In this mode, the CPU/DMA is configured as a 32-bit access mode, and the data is stored in the DATAIN0[15:0] bit domain of the lower 16 bits and the DATAIN1[15:0] bit domain of the higher 16 bits. When writing 32-bit data once, the channel filter must perform two input samples to clear the HPDF_CHxPDI register. The channel filter samples the DATAIN0[15:0] bit domain for the first time and the DATAIN1[15:0] bit domain for the second time.

3. Dual channel mode (DPM[1:0] = 2'b10):

In this mode, the CPU/DMA is configured as a 32-bit access mode, and the data is stored in the DATAIN0[15:0] bit domain of the lower 16 bits and the DATAIN1[15:0] bit domain of the higher 16 bits. The data in the DATAIN0[15:0] bit field is used for the current channel x, and the data in the DATAIN1[15:0] bit field is automatically copied to the lower 16 bits of the parallel data input register of the channel x+1, and the data is used for the channel x+1. CPU/DMA writes data once, digital filter performs two sampling, the first is channel x sampling, the second is channel x+1 sampling.

In HPDF module, only even channel (channel0) supports dual channel mode. If odd channel (channel1) is configured as dual channel mode, the parallel data input register HPDF_CHxPDI of this channel is write protected. If channel x is even and configured as dual channel mode, odd channel x+1 must be configured as standard mode.

The operation mode of HPDF_CHxPDI register is as follows:

Table 30-3. Parallel data packed mode

Channel	Packed mode					
	Standard mode		Interleaving mode		Dual channel mode	
	DATAIN1	DATAIN0	DATAIN1	DATAIN0	DATAIN1	DATAIN0
Channel0	Write protect	CH0 sampling	CH0 second sampling	CH0 first sampling	CH1 sampling	CH0 sampling
Channel1	Write protect	CH1 sampling	CH1 second sampling	CH1 first sampling	Write protect	Write protect

CPU/DMA should write to HPDF_CHxPDI register after the channel is enabled, because after the channel is enabled, the channel conversion will be started, and the data in HPDF_CHxPDI register will be discarded before the channel conversion is started.

30.3.6. Regular group conversion

HPDF module has two multiplexing channels, which can be used for regular group conversion or inserted group conversion respectively. If the channel is disabled (CHEN = 0), enabling the channel conversion will cause the channel to remain in the conversion state. The channel can be restored only by enabling the channel (CHEN=1) or disabling the HPDF module (HPDFEN=0).

The regular group selects only one of the two channels, which is determined by the RCS bit in the HPDF_FLTyCTL0 register. At the same time, only one regular conversion can be executed or pending. If an existing regular conversion request has not been completed, the new regular conversion start request is ignored. The priority of regular conversion is lower than that of inserted group conversion and can be interrupted by inserted group conversion request.

The conversion time of regular group: $t = \text{CTCNT}[27:0] * f_{\text{HPDFCLK}}$.

Conversion start mode

Regular group conversion can only be achieved by software startup. There are two modes of software startup, the specific methods are as follows:

1. General software startup: write 1 to SRCs bit in HPDF_FLTyCTL0 register.
2. Software synchronous start: Set the RCSYN bit in HPDF_FLTyCTL0 register and start the regular conversion of HPDF_FLT0 by general software startup. Then HPDF_FLTy also starts the regular conversion synchronously.

Conversion mode

Regular group transformation supports continuous mode and fast mode.

Continuous mode

Set the RCCM bit to 1 in HPDF_FLTyCTL0 register to enable continuous mode. In continuous mode, after the software starts the regular group conversion, the conversion regular group channel conversion is repeated. When the RCCM bit is cleared, the regular conversion in continuous mode stops immediately.

Fast mode

Enable fast mode by setting FAST bit to 1 in HPDF_FLTyCTL0. In fast mode, it can improve the data rate in continuous mode. Because in continuous mode, if the data is continuously converted from one channel, there is no need to fill the filter with new data, because the data in the filter is valid data sampled from the previous continuous mode. The increase in data rate is determined by the order of the selected filter.

After the continuous conversion is started, the time for the first conversion of the fast mode to the non open fast mode is the same, and then the subsequent conversion will be completed at a shorter time interval.

30.3.7. Inserted group conversion

The conversion channel of the inserted group must select at least one of the two channels. You can select which channel to convert into the inserted group by the ICGSEL[1:0] bit field in the HPDF_FLTyICGS register. ICGSEL[y]=1 means channel x is the inserted group channel.

The priority of the inserted group is higher than that of the regular group. The ongoing regular group conversion will be interrupted by the inserted group conversion request. Wait for the inserted group to complete the conversion and restart the interrupted regular conversion. At the same time, only one inserted conversion is in execution or pending state. If an existing inserted conversion request has not been completed, the new inserted conversion start request will be ignored.

The conversion time of the inserted group $t = \text{CTCNT}[27:0] * f_{\text{HPDFCLK}}$.

Conversion start mode

The conversion of the inserted group can be achieved through channel software startup and trigger startup.

1. General software startup: write 1 to the SICC bit in HPDF_FLTyCTL0 register.
2. Software synchronous startup: Set the ICSYN bit in HPDF_FLTyCTL0 register to start synchronously. When using general software to start the inserted group conversion of HPDF_FLT0, the channel 1 that enables the synchronous start function also starts the inserted conversion.
3. Trigger startup: When the ICTSSEL[4:0] bit field in the HPDF_FLTyCTL0 register is written with a value other than 0, it indicates that trigger start is enabled and trigger signal source is selected at the same time. The effective edge of the trigger is determined by the ICTEEN[1:0] bit field.

The trigger signals of the inserted group are shown in the following table:

Table 30-4. Trigger signal of inserted group

Trigger signal	Type	Signal source
HPDF_ITRG0	internal signal	TIMER1_TRGO
HPDF_ITRG1	internal signal	TIMER2_TRGO
HPDF_ITRG2	internal signal	TIMER3_TRGO
HPDF_ITRG3	internal signal	TIMER4_TRGO
HPDF_ITRG[4~23]	reserve	reserve
HPDF_ITRG24	external signal	EXTI11
HPDF_ITRG25	external signal	EXTI15
HPDF_ITRG26	external signal	TIMER5_TRGO
HPDF_ITRG[27~31]	reserve	reserve

Scan conversion mode

By setting the SCMOD bit in HPDF_FLTyCTL0 register, the scan conversion mode for inserted group conversion can be enabled. In the scan mode, when the inserted group conversion is triggered, all channels in the inserted group will be converted sequentially starting from the lowest channel.

If the scan mode is disabled, each time the inserted group conversion is triggered, only one

channel in the inserted group will be converted, and the next trigger will select another channel. And writing to the ICGSEL[1:0] bit field will use the lowest channel as the selected conversion channel.

Conversion request priority

The conversion of the inserted group has a higher priority than the regular group. The regular conversion that is already in progress will be immediately interrupted by the inserted conversion request. When the inserted conversion sequence ends, if RCCM remains at 1, the continuous regular conversion will start again. The value of the RCHPDT bit indicates that the interrupted regular conversion is delayed.

If an inserted conversion is pending or already in progress, you cannot start other inserted conversions: as long as ICPF=1, any request to launch an inserted conversion (software or trigger start) will be ignored. The regular conversion is the same.

When the inserted conversion is in progress (ICPF=1), write 1 to the SRCS bit of HPDF_FLTyCTL0 to request regular conversion. When the inserted sequence is completed, the priority indicates the next step to perform regular conversion, and the delayed start is indicated by the RCHPDT bit.

30.3.8. Digital filter

The digital filter of the HPDF module is of Sinc^X type. The input data stream is filtered by Sinc^X, thereby reducing the output data rate and increasing the output data resolution. Configure the order and oversampling rate (decimation filtering) of the Sinc^X filter by the SFO[2:0] and SFOR[9:0] bits in the HPDF_FLTySFCFG register. The user can configure the order and oversampling rate of the Sinc^X filter according to the desired resolution. The relationship between the maximum output resolution of Sinc^X filtering and oversampling filtering is as follows:

Table 30-5. The relationship between the maximum output resolution and oversampling filtering of Sinc^X filtering

SFOR	Sinc	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	$\pm x$	$\pm x^2$	$\pm 2x^2$	$\pm x^3$	$\pm x^4$	$\pm x^5$
4	± 4	± 16	± 32	± 64	± 256	± 1024
8	± 8	± 64	± 64	± 512	± 4096	± 32768
32	± 32	± 1024	± 2048	± 32768	± 1048576	± 33554432
64	± 64	± 4096	± 8192	± 262144	± 16777216	± 1073741824
128	± 128	± 16384	± 32768	± 2097152	± 268435456	-
256	± 256	± 65536	± 131072	± 16777216	Under full-scale input conditions, the result will overflow	
1024	± 1024	± 1048576	± 2097152	± 1073741824		

Note: The maximum output resolution in this table comes from the peak data value of the filter output.

30.3.9. Integrator

The integrator performs further oversampling rate (decimation rate) and resolution improvement on the data from the digital filter. The integrator performs a simple summation operation on a given number of data samples from the filter. The output data of the integrator comes from the sum of the output samples of the filter, and the number of output samples is determined by the oversampling rate of the integration. The oversampling rate (decimation filtering) of the integrator can be configured by IOR[7:0] in HPDF_FLTySFCFG register. The relationship between the maximum output resolution, oversampling rate, and Sinc filter order of the integrator is as follows:

Table 30-6. Relationship between the maximum output resolution and IOR, SFOR, SFO of the integrator

Filter type	Integrator maximum output resolution
Sinc	$\pm(SFOR \times IOR)$
$Sinc^2$	$\pm(SFOR^2 \times IOR)$
FastSinc	$\pm(2SFOR^2 \times IOR)$
$Sinc^3$	$\pm(SFOR^3 \times IOR)$
$Sinc^4$	$\pm(SFOR^4 \times IOR)$
$Sinc^5$	$\pm(SFOR^5 \times IOR)$

30.3.10. Threshold monitor

The threshold monitor of the HPDF module is used to monitor the serial input data of the channel or the final output data after the channel conversion. When the data reaches the threshold set by the threshold monitor (maximum or minimum threshold), an interrupt or break event will be generated. The maximum threshold is determined by the HTVAL[23:0] bits in HPDF_FLTyTMHT register, and the minimum threshold is determined by the LTVVAL[23:0] bits in HPDF_FLTyTMLT register.

The HPDF module has two threshold monitor. By configuring the TMCHEN[1:0] bit field in HPDF_FLTyCTL1 register, it determines whether the analog threshold monitor x monitors the input channel. For example, TMCHEN[1]=1 in HPDF_FLT0CTL1 register means threshold monitor 0 monitors channel 1.

Threshold monitor working mode

The working mode of the threshold monitor is divided into standard mode and fast mode. Fast mode is to configure the serial input data of the threshold monitor monitoring channel and compare it with the threshold. Standard mode is to configure the final data output after the threshold monitor monitor channel conversion (stored in the inserted group data register HPDF_FLTyIDATA or the regular group data register HPDF_FLTyRDATA). The fast mode of the threshold monitor can be enabled by the TMFM bit in HPDF_FLTyCTL0. The

characteristics in both cases are as follows:

Table 30-7. Features of threshold monitor working mode

Mode	Enable Bit	Channel Data Source	Analog Input Data Source	Input Data Resolution	Detailed Description
Standard mode	TMFM=0	Serial data stream, Parallel data	HPDF final data output	24bit	The threshold monitor monitors the final data output after the channel conversion. Slow response time, not suitable for overcurrent/overvoltage detection.
Fast mode	TMFM=1	Serial data stream	Serial data stream	16bit	The input data is provided in continuous mode, and the threshold monitor directly monitors the serial input data, regardless of rules or injection conversion. Fast response time, suitable for overcurrent/overvoltage detection.

In fast mode, the threshold monitor uses only the upper 16 bits of the threshold (maximum threshold HTVAL[23:0] or minimum threshold LTVL[23:0]) to compare with the serial input data of the channel, that is, only the upper 16 bits of HTVAL[23:0] and LTVL[23:0] define the threshold, because the resolution of the threshold monitor filter is 16 bits.

In non-fast mode of threshold monitor, the final data of right shift and offset calibration will be compared with HTVAL[23:0] and LTVL[23:0].

Threshold monitor fast mode

In fast mode, the filter of the threshold monitor will be used, and the oversampling rate (decimation rate) and order of the threshold monitor filter can be set in HPDF_CHxCFG1 register.

The configuration of the threshold monitor is flexible. A threshold monitor can be configured to monitor multiple channels by the TMCHEN[1:0] bit field in HPDF_FLTyCTL1 register. In this case, when multiple channels send out requests, the threshold monitor will preferentially process requests with small channel numbers, and then process requests with large channel numbers. Each threshold monitor has a status register HPDF_FLTyTMSTAT. When the monitored channel exceeds the threshold, the corresponding flag in the HTF[1:0] or LTF[1:0] bit field will be set. If HTF[0]=2b'01, it means that channel 0 occurred an event that exceeds

the upper threshold.

After each channel sends a comparison request, it will be executed within 8 HPDF clock cycles. Therefore, the bandwidth of each channel is limited to 8 HPDF clock cycles (if TMCHEN[1:0]=3). Since the maximum sampling frequency of the input channel is $f_{\text{HPDFCLK}}/4$, at this input clock speed, the threshold monitor filter cannot be bypassed (TMFOR=0). Therefore, the user must correctly configure the threshold monitor filter parameters and the number of channels monitored based on the input sampling clock speed and f_{HPDFCLK} .

In fast mode, reading the TMDATA[15:0] bit field in HPDF_CHxTMFDT register to get the threshold monitor filter data for the given channel x. The number of serial samples required for a result of the threshold monitor filter output (at the serial input clock frequency f_{CKIN}) is as follows:

1. First conversion:

FastSinc filter: Number of samples is equal to $(TMSFO \times 4 + 2) + 1$.

$Sinc^x$ filter ($x=1..5$): Number of samples is equal to $((TMSFO[4:0] + 1) \times TMFOR) + TMSFO + 1$.

2. Subsequent conversions other than the first conversion:

FastSinc and $Sinc^x$ filter ($x=1..5$): Number of samples is equal to $(TMSFO[4:0] + 1) \times (IOR[7:0] + 1)$.

Threshold monitor flag

The global state of the threshold monitor is the TMEOF flag in HPDF_FLTySTAT register. When TMEOF=1, it indicates that at least one threshold monitor event has occurred, that is, an event that exceeds the (upper/lower limit) threshold is generated. If the threshold monitor event interrupt TMIE=1 in HPDF_FLTyCTL1 register is enabled, an threshold monitor interrupt can be generated. When all HTF[1:0] and LTF[1:0] are cleared, the TMEOF bit is cleared.

The HPDF_FLTyTMSTAT register defines the error event flag of the channel exceeding the threshold. The HTF[1:0] bit field indicates whether the maximum threshold HTVAL[23:0] has been exceeded on the channel x. The LTF[1:0] bit field indicates whether the minimum threshold LTVL[23:0] value has been exceeded on channel x. Clear the threshold event flag by writing "1" to the corresponding HTFC[1:0] or LTFC[1:0] bit in the HPDF_FLTyTMFC register.

There are 2 break output signals HPDF_BREAK[0] and HPDF_BREAK[1] in the HPDF module. The break output signals are assigned to threshold monitor threshold event by setting the HTBSD[1:0] and LTBSD[1:0] bit fields in the HPDF_FLTyTMHT register and the HPDF_FLTyTMLT register.

The signal source of the broken output signal HPDF_BREAK[0] is TIMER15, and the signal source of HPDF_BREAK[1] is TIMER16.

30.3.11. Malfunction monitor

The purpose of the malfunction monitor is to be able to send a signal with an extremely fast response time when the analog signal reaches a saturation value and remains in this state for a given time. This feature can be used to detect short-circuit or open-circuit faults (e.g. overcurrent/overvoltage). The broken output signals HPDF_BREAK[0] and HPDF_BREAK[1] can be assigned to the malfunction monitor event, which can be configured by the MMBSD[1:0] bit field in the HPDF_CHxCFG1 register. The broken output signal is the same as the threshold monitor.

The input data of the malfunction monitor comes from the serial input data of the channel. When the channel input data source is parallel data, the malfunction monitor function is prohibited. There is an up counter on each input channel to record how many consecutive 0 or 1 on the output of the serial data receiver. When the counter reaches the threshold value of the malfunction monitor (MMCT[7:0] bits in the HPDF_CHxCFG1 register), a malfunction event occurs. If a 0-1 or 1-0 change is encountered when monitoring the data stream, the value of the counter will be automatically cleared and counted again.

The user can enable the malfunction monitor function by setting the MMEN bit in HPDF_CHxCTL register. When a malfunction event occurs on the channel, the corresponding malfunction monitor flag MMF[1:0] is set. The corresponding flag can be cleared by MMFC[1:0] in HPDF_FLTyINTC. If channel x is disabled (CHEN=0), the hardware will also clear the malfunction monitor flag.

30.3.12. Extremes monitor

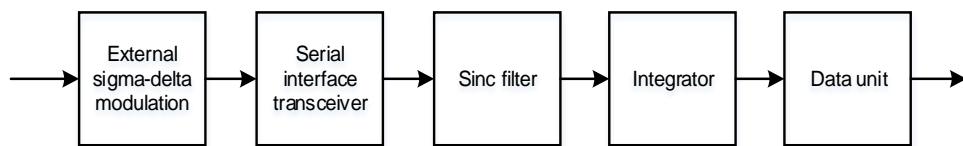
The extremes monitor is used to sample the minimum and maximum values (peak to peak) of the final output data word. An extremes monitor can be configured to sample the extreme values of multiple channels through the EMCS[1:0] bit field in HPDF_FLTyCTL1 register.

If the sampling final output data word is higher than the value in the maximum value register of the extremes monitor (MAXVAL[23:0] bits in the HPDF_FLTyEMMAX register), the value of this register is updated to the current final output data. If the sampling final output data word is smaller than the value in the minimum value register of the extremes monitor (MINVAL[23:0] bits in HPDF_FLTyEMMIN register), the value of this register is updated to the current final output data. The values of the MAXDC bit and the MINDC bit indicate which channel the maximum/minimum value comes from.

When reading the HPDF_FLTyEMMAX or HPDF_FLTyEMMIN register, the maximum or minimum value is updated with the reset value.

30.3.13. Data unit

The data unit is the last part of data processing in the entire HPDF module, and the flow of data processing by the HPDF module is shown in the following figure.

Figure 30-7. HPDF module external input data processing flow


The output data rate depends on the serial data stream rate, filter and integrator settings. The maximum output data rate is shown in the table below.

Table 30-8. Maximum output rate

Input source	Conversion mode	Filter type	Maximum output data rate (samples/second)
Serial input	Non-fast mode (FAST=0)	Sinc ^X	$\frac{f_{CKIN}}{SFOR \times (IOR-1+SFO)+(SFO+1)}$
	Non-fast mode (FAST=0)	FastSinc	$\frac{f_{CKIN}}{SFOR \times (IOR-1+4)+(2+1)}$
	Fast mode (FAST=1)	FastSinc and Sinc ^X	$\frac{f_{CKIN}}{SFOR \times IOR}$
Parallel input	Non-fast mode (FAST=0)	Sinc ^X	$\frac{f_{DATA}}{SFOR \times (IOR-1+SFO)+(SFO+1)}$
	Non-fast mode (FAST=0)	FastSinc	$\frac{f_{DATA}}{SFOR \times (IOR-1+4)+(2+1)}$
	Fast mode (FAST=1)	FastSinc and Sinc ^X	$\frac{f_{DATA}}{SFOR \times IOR}$

Note: f_{DATA} is the parallel data rate of the CPU/DMA input. When the filter is bypassed, $f_{DATA} \leq f_{HPDFCLK}$ must be satisfied.

Signed data format

Signed data in HPDF module: parallel data register, regular and inserted group data register, threshold monitor value, extreme monitor value, and offset calibration are all signed formats. The most significant bit of the output data indicates the sign of the value, and the data is in two's complement format.

Since all operations in digital processing are performed on 32-bit signed registers, the following conditions must be met in order for the result not to overflow:

- When using Sinc^X filter ($x=1..5$): $(SFOR^{SFO}) \times IOR \leq 2^{31}$.
- When using FastSinc filter: $2 \times (SFOR^2) \times IOR \leq 2^{31}$.

Data right bit shift

Since the final data width is 24 bits and the data from the processing path can be up to 32 bits, the right shift of the final data is performed in this module. For each selected input channel, the number of bits shifted to the right can be configured in the DTRS[4:0] bit field in

the HPDF_CHxCFG0 register. The result is round to the nearest value by abandoning the lowest bit.

Data offset calibration

In the HPDF module, each channel has a data offset calibration value, which is stored in the CALOFF[23:0] bit field in HPDF_CHxCFG0 register. When offset calibration is performed, the offset calibration value is subtracted from the output data of the channel to obtain the final data output by the HPDF module.

Data offset calibration occurs after the right shift of the data.

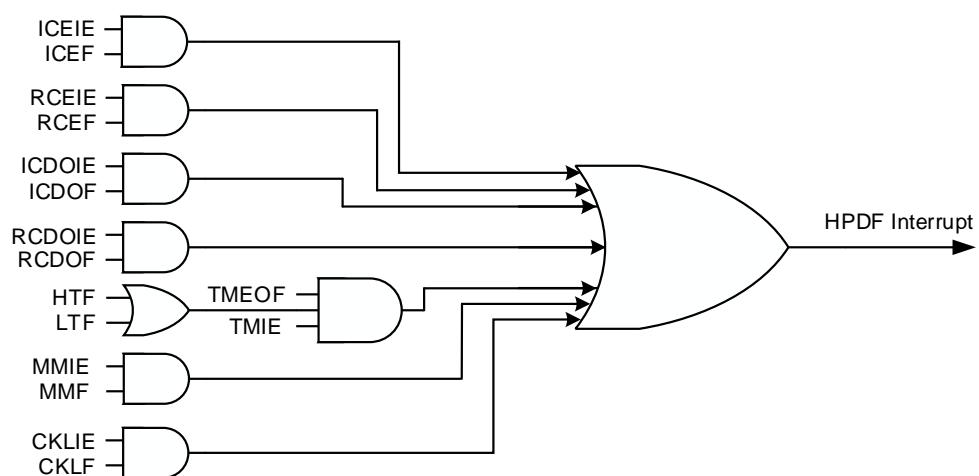
30.3.14. HPDF interrupt

HPDF interrupt events can be divided into channel conversion interrupt events, threshold monitor interrupt events, malfunction monitor interrupt events, and channel clock loss interrupt events. The specific interrupt event description is as [Table 30-9. HPDF interrupt event](#) shown.

Table 30-9. HPDF interrupt event

Interrupt event	description	Clear	Enable interrupt
ICEF	end of inserted conversion	Read HPDF_FLTyIDATA register	ICEIE
RCEF	end of regular conversion	Read HPDF_FLTyRDATA register	RCEIE
ICDOF	inserted conversion data overflow	Write 1 to the ICDOFC bit	ICDOIE
RCDOF	regular conversion data overflow	Write 1 to RCDOFC bit	RCDOIE
TMEOF HTF[1:0] LTF[1:0]	threshold monitor events	Write 1 to HTFC[1:0] bit field Write 1 to LTFC[1:0] bit field	TMIE
MMF	malfunction event	Write 1 to MMFC[1:0] bits	MMIE
CKLF	Channel clock loss event	Write 1 to CKLFC[1:0] bits	CKLIE

HPDF interrupt logic is as [Figure 30-8. HPDF interrupt logic diagram](#) shown.

Figure 30-8. HPDF interrupt logic diagram

30.4. Register definition

HPDF secure access base address: 0x5001 6000
 HPDF non-secure access base address: 0x4001 6000

30.4.1. HPDF channel x registers (x=0, 1)

Channel x control register (HPDF_CHxCTL)

Address offset: 0x00 + 0x20 * x, (x = 0, 1)

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
HPDFEN	CKOUTS EL	CKOUTD M	Reserved				CKOUTDIV[7:0]								
rw	rw														rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM[1:0]	CMSD[1:0]	Reserved			CHPINSE L	CHEN	CKLEN	MMEN	Reserved	SPICKSS[1:0]	SITYP[1:0]				
rw	rw				rw	rw	rw	rw	rw	rw	rw				rw

Bits	Fields	Descriptions
31	HPDFEN	Global enable for HPDF interface 0: HPDF disabled 1: HPDF enabled If HPDFEN=0, the HPDF_FLTySTAT register and HPDF_FLTyTMSTAT register is set to reset state. This bit is only available in HPDF_CH0CTL.
30	CKOUTSEL	Serial clock output source selection 0: Serial clock output source is from CK_HPDF clock 1: Serial clock output source is from CK_HPDFAUDIO clock This bit can be configured only when HPDFEN=0. This bit is only available in HPDF_CH0CTL.
29	CKOUTDM	Serial clock output duty mode 0: Serial clock output duty mode disable 1: Serial clock output duty mode enable, the duty is 1:1 This bit can be configured only when HPDFEN=0. This bit is only available in HPDF_CH0CTL.
28:24	Reserved	Must be kept at reset value.
23:16	CKOUTDIV[7:0]	Serial clock output divider

0: Output clock generation is disabled (CKOUT signal is set to low state)
 1~255: The value of division for the serial clock output is CKOUTDIV+1.
 CKOUTDIV also defines the threshold for a clock loss detection.
 This value can only be modified when HPDFEN=0. During a HPDF clock cycle after
 HPDFEN=0, the CKOUT is set to low state.
 This bit is only available in HPDF_CH0CTL.

15:14	DPM[1:0]	Data packing mode for HPDF_CHxPDI register 00: Standard mode 01: Interleaved mode 10: Dual mode 11: Reserved For a detailed introduction of data encapsulation mode, please refer to <u>Parallel data packed mode.</u> These bits can be configured only when CHEN=0.
13:12	CMSD[1:0]	Channel x multiplexer select input data source 00: Input data source for channel x is taken from serial inputs 01: Reserved 10: Input data source for channel x is taken from internal HPDF_CHxPDI register 11: Reserved The HPDF_CHxPDI register is write protected when these bits are reset. These bits can be configured only when CHEN=0.
11:9	Reserved	Must be kept at reset value.
8	CHPINSEL	Channel inputs pins selection 0: Channel inputs select pins of the current channel x 1: Channel inputs select pins of the next channel. This bit can be configured only when CHEN=0.
7	CHEN	Channel x enable 0: Channel x disabled 1: Channel x enabled If channel x is enabled, then serial data will be received based on the given channel settings.
6	CKLEN	Clock loss detector enable 0: Clock loss detector disabled 1: Clock loss detector enabled
5	MMEN	Malfunction monitor enable 0: malfunction monitor is no effect 1: malfunction monitor is effect
4	Reserved	Must be kept at reset value.
3:2	SPICKSS[1:0]	SPI clock source select

00:External CKIN_x input is selected for SPI clock source, sampling point is determined by SITYP[1:0]

01: Internal CKOUT output is selected for SPI clock source, sampling point is determined by SITYP[1:0]

10: Internal CKOUT is selected for SPI clock source, sampling point on each second CKOUT falling edge.

11: Internal CKOUT output is selected for SPI clock source, sampling point on each second CKOUT rising edge.

These bits can be configured only when CHEN=0.

1:0	SITYP[1:0]	Serial interface type
		00: SPI interface, sample data on rising edge
		01: SPI interface, sample data on falling edge
		10: Manchester coded input: rising edge = logic 0, falling edge = logic 1
		11: Manchester coded input: rising edge = logic 1, falling edge = logic 0

These bits can only be configured when CHEN=0.

Channel x configuration register 0 (HPDF_CHxCFG0)

Address offset: 0x04 + 0x20 * x, (x = 0, 1)

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
CALOFF[23:8]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALOFF[7:0]								DTRS[4:0]				Reserved			
rw								rw							

Bits	Fields	Descriptions
31:8	CALOFF[23:0]	24-bit calibration offset Calibration offset must be performed for each conversion result of the channel. These bits can be set by software.
7:3	DTRS[4:0]	Data right bit-shift 0-31: The number of bits that determine the right shift of data Bit-shift is performed before offset correction. The data shift rounds the result to the nearest integer value and the sign is preserved. These bits can be configured only when CHEN=0 (in HPDF_CHx CTL register).
2:0	Reserved	Must be kept at reset value.

Channel x configuration register 1 (HPDF_CHxCFG1)

Address offset: $0x08 + 0x20 * x$, ($x = 0, 1$)

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:22	TMSFO[1:0]	Threshold monitor Sinc filter order selection 00: FastSinc filter 01: Sinc ¹ filter 10: Sinc ² filter 11: Sinc ³ filter These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
21	Reserved	Must be kept at reset value
20:16	TMFOR[4:0]	Threshold monitor filter oversampling rate (decimation rate) 0 - 31: The filter decimation rate equal to TMFOR[4:0]+ 1 If TMFOR=0, the filter is bypassed. These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
15:14	Reserved	Must be kept at reset value
13:12	MMBSD[1:0]	Malfunction monitor break signal distribution 00: Break signal not is distributed to malfunction monitor on channel 01: Break signal 0 is distributed to malfunction monitor on channel x 10: Break signal 1 is distributed to malfunction monitor on channel x 11: Break signal 0 and 1 is distributed to malfunction monitor on channel x
11:8	Reserved	Must be kept at reset value.
7:0	MMCT[7:0]	Malfunction monitor counter threshold These bits be used determine the count value of malfunction monitor counter threshold. The count value is written by software. If the count value is reached, then an event of malfunction monitor occurs on a given channel.

Channel x threshold monitor filter data register (HPDF_CHxTMFDT)

Address offset: $0x0C + 0x20 * x$, ($x = 0, 1$)

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
TMDATA[15:0]															

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TMDATA[15:0]	Threshold monitor data The data is come from the threshold monitor filter and continuously converted (no trigger) for this channel.

Channel x parallel data input register (HPDF_CHxPDI)

Address offset: $0x10 + 0x20 * x$, ($x = 0, 1$)

Reset value: 0x0000 0000

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

This register contains 16-bit input data to be processed by HPDF filter module.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN1[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN0[15:0]															

rw

Bits	Fields	Descriptions
31:16	DATAIN1[15:0]	Data input for channel x or channel x+1 Data can be written by CPU/DMA. If DPM[1:0]=0 (standard mode), DATAIN1[15:0] is write protected. If DPM[1:0]=1 (interleaved mode), second channel x data sample is stored into DATAIN1[15:0]. First channel x data sample is stored into DATAIN0[15:0]. Both samples are read sequentially by HPDF_FLT _y filter. If DPM[1:0]=2 (dual mode): For channel 0: sample in DATAIN1[15:0] is automatically copied into DATAIN0[15:0] of channel 1.

For channel 1: DATAIN1[15:0] is write protected.

The more details refer to [错误!未找到引用源。](#)

DATAIN1[15:0] is a signed format data.

15:0	DATAIN0[15:0]	<p>Data input for channel x</p> <p>Data can be written by CPU/DMA.</p> <p>If DPM[1:0]=0 (standard mode), channel x data sample is stored into DATAIN0[15:0].</p> <p>If DPM[1:0]=1 (interleaved mode), first channel x data sample is stored into DATAIN0[15:0]. Second channel x data sample is stored into DATAIN1[15:0]. Both samples are read sequentially by HPDF_FLTy filter.</p> <p>If DPM[1:0]=2 (dual mode):</p> <p>For channel 0: Channel x data sample is stored into DATAIN0[15:0].</p> <p>For channel 1: DATAIN0[15:0] is write protected.</p> <p>The more details refer to 错误!未找到引用源。</p> <p>DATAIN0[15:0] is a signed format data.</p>
------	---------------	--

Channel x pulse skip register (HPDF_CHxPS)

Address offset: 0x14 + 0x20 * x, (x = 0, 1)

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										PLSK[5:0]					
rw															

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	PLSK[5:0]	<p>Pulses to skip for input data skipping function</p> <p>0-63: Defines the number of serial input samples that will be skipped.</p> <p>Skipping function is take effect immediately after writing to this field. Read PLSK[5:0] to return the remaining value of the pulses which will be skipped.</p> <p>The value of PLSK[5:0] can be updated even PLSK[5:0] is not zero.</p>

30.4.2. HPDF filter y registers (y=0, 1)

Filter y control register 0 (HPDF_FLTyCTL0)

Address offset: 0x100 + 0x80 * y, (y = 0, 1)

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	TMFM	FAST	Reserved			RCS	Reserved		RCDMAE N	Reserved	RCSYN	RCCM	SRCS	Reserved	
	rw	rw				rw			rw	rw	rw	rw	rw	rt_w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ICTEEN[1:0]		ICTSSEL[4:0]				Reserved		ICDMAE N	SCMOD	ICSYN	Reserved	SICC	FLTEN	
	rw								rw	rw	rw	rw	rt_w		rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	TMFM	Threshold monitor fast mode 0: Threshold monitor w atch on the final data after performing offset correction and right shift 1: Threshold monitor w atch on serial input data stream
29	FAST	Fast conversion mode for regular conversions 0: Fast conversion mode disabled 1: Fast conversion mode enabled If fast mode is enabled, the normal conversion in continuous mode (except for the first conversion) is performed faster than the conversion in standard mode. This bit has no effect on conversions w hich are not continuous. This bit can be configured only w hen FLTEN=0.
28:25	Reserved	Must be kept at reset value
24	RCS	Regular conversion channel selection 0: Channel 0 is selected as the regular conversion channel 1: Channel 1 is selected as the regular conversion channel When RCPF=1, writing this bit takes effect w hen the next regular conversion begins.
23:22	Reserved	Must be kept at reset value
21	RCDMA EN	DMA channel enabled to read data for the regular conversion 0: Disable the DMA channel to read regular data 1: Enable the DMA channel to read regular data This bit can be configured only w hen FLTEN=0.
20	Reserved	Must be kept at reset value.
19	RCSYN	Regular conversion synchronously w ith HPDF_FLT0 0: Do not launch a regular conversion synchronously w ith HPDF_FLT0 1: Launch a regular conversion synchronously in HPDF_FLT0 w hen a regular

conversion is launched in HPDF_FLT0
 If RCSYN=1 in HPDF_FLT0CTL0 register, the regular conversion channel will be Launched synchronously which selected in HPDF_FLTy CTL0.
 This bit can be configured only when FLTN=0.

18	RCCM	Regular conversions continuous mode 0: The regular channel is converted just once for each conversion request 1: The regular channel is converted repeatedly after each conversion request Writing "0" to this bit will immediately stop continuous mode during a continuous regular conversion.
17	SRCS	Start regular channel conversion by software 0: No effect 1: Make a request to start regular channel conversion If RCPF=1, invalid write to SRCS, and if RCSYN=1, write '1' to SRCS, launch a regular conversion synchronously. This bit is always read as '0'.
16:15	Reserved	Must be kept at reset value.
14:13	ICTEEN[1:0]	Inserted conversions trigger edge enable 00: Disable trigger detection 01: Each rising edge on the trigger signal makes a request to start an inserted conversion 10: Each falling edge on the trigger signal makes a request to start an inserted conversion 11: The edge (rising edges and falling edges) on the trigger signal make requests to start inserted conversions This bit can be configured only when FLTN=0.
12:8	ICTSSEL[4:0]	Inserted conversions trigger signal selection 0x0~0x1F: The value indicates that different trigger signals are selected to start the conversion 0x00: HPDF_ITRG0 (TIM1_TRGO) is selected to start inserted conversion 0x01: HPDF_ITRG1 (TIM2_TRGO) is selected to start inserted conversion 0x02: HPDF_ITRG2 (TIM3_TRGO) is selected to start inserted conversion 0x03: HPDF_ITRG3 (TIM4_TRGO) is selected to start inserted conversion 0x04~0x17: Reserved 0x18: HPDF_ITRG24 (EXTI11) is selected to start inserted conversion 0x19: HPDF_ITRG25 (EXTI15) is selected to start inserted conversion 0x1A: HPDF_ITRG25 (TIM5_TRGO) is selected to start inserted conversion 0x1B~0x1F: Reserved The maximum delay from the generation of trigger signal to the start of synchronous trigger is 1 $f_{HPDFCLK}$ clock cycle, and the delay of asynchronous trigger is 2-3 $f_{HPDFCLK}$ clock cycles. This bit can be configured only when FLTN=0.

7:6	Reserved	Must be kept at reset value.
5	ICDMA EN	DMA channel enabled to read data for the inserted channel group 0: Disable DMA channel to read inserted conversions data 1: Enable DMA channel to read inserted conversions data This bit can be configured only when FLTEN=0.
4	SCMOD	Scan conversion mode of inserted conversions 0: One channel conversion is performed from the inserted channel group and next the channel is selected from this group. 1: The series of conversions for the inserted group channels is executed, starting over with the lowest selected channel. If SCMOD=0, writing ICGSEL will resets the channel selection to the lowest selected channel. This bit can be configured only when FLTEN=0.
3	ICSYN	Inserted conversion synchronously with the HPDF_FLT0 SICC trigger 0: Do not launch an inserted conversion synchronously with HPDF_FLT0 1: Launch an inserted conversion synchronously in HPDF_FLTyCTL0 when an inserted conversion is launched by SICC trigger in HPDF_FLT0CTL0. This bit can be configured only when FLTEN=0.
2	Reserved	Must be kept at reset value
1	SICC	Start inserted group channel conversion 0: No effect. 1: Makes a request to convert the channels in the inserted conversion group. If ICPF=1 already, invalid write to SICC. If RCSYN=1, write '1' to SICC, launch an inserted conversion synchronously. This bit is always read as '0'.
0	FLTEN	HPDF_FLTy enable 0: HPDF_FLTy is disabled. 1: HPDF_FLTy is enabled. If HPDF_FLTy is enabled, then HPDF_FLTy starts operating according to its setting. If HPDF_FLTy is disabled, all conversions of given HPDF_FLTy are stopped immediately and all HPDF_FLTy functions are stopped. Meanwhile HPDF_FLTySTAT register and HPDF_FLTyTMSTAT register is set to the reset state.

Filtery control register 1 (HPDF_FLTyCTL1)

Address offset: 0x104 + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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

Reserved															TMCHEN[1:0]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					EMCS[1:0]	Reserved	CKLIE	MMIE	TMIE	RDOVRI E	IDOVRI E	RCEIE	ICEIE		

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	TMCHEN[1:0]	<p>Threshold monitor channel enable</p> <p>These bits select the input channel to be guarded continuously by the threshold monitor.</p> <ul style="list-style-type: none"> 00: Threshold monitor y is disabled on channel 0 and channel 1 01: Threshold monitor y is enabled on channel 0 10: Threshold monitor y is enabled on channel 1 11: Threshold monitor y is enabled on channel 0 and channel 1
15:10	Reserved	Must be kept at reset value
9:8	EMCS[1:0]	<p>Extremes monitor channel selection</p> <p>These bits select the input channels to be taken by the extremes monitor.</p> <ul style="list-style-type: none"> 00: Extremes monitor y does not monitor data from channel 0 and channel 1 01: Extremes monitor y monitor data from channel 0 10: Extremes monitor y monitor data from channel 1 11: Extremes monitor y monitor data from channel 0 and channel 1
7	Reserved	Must be kept at reset value
6	CKLIE	<p>Clock loss interrupt enable</p> <p>0: Detection of channel input clock loss interrupt is disabled</p> <p>1: Detection of channel input clock loss interrupt is enabled</p> <p>This bit is only available in HPDF_FLT0CTL1 register.</p>
5	MMIE	<p>Malfunction monitor interrupt enable</p> <p>0: malfunction monitor interrupt is disabled</p> <p>1: malfunction monitor interrupt is enabled</p> <p>This bit is only available in HPDF_FLT0CTL1 register.</p>
4	TMIE	<p>Threshold monitor interrupt enable</p> <p>0: Threshold monitor interrupt is disabled</p> <p>1: Threshold monitor interrupt is enabled</p>
3	RCDOIE	<p>Regular conversion data overflow interrupt enable</p> <p>0: Regular conversion data overflow interrupt is disabled</p> <p>1: Regular conversion data overflow interrupt is enabled</p>
2	ICDOIE	Inserted conversion data overflow interrupt enable

		0: Inserted data overflow interrupt is disabled 1: Inserted data overflow interrupt is enabled
1	RCEIE	Regular conversion end interrupt enable 0: Regular conversion end interrupt is disabled 1: Regular conversion end interrupt is enabled
0	ICEIE	Inserted conversion end interrupt enable 0: Inserted conversion end interrupt is disabled 1: Inserted conversion end interrupt is enabled

Filter y status register (HPDF_FLTySTAT)

Address offset: $0x108 + 0x80 * y$, ($y = 0, 1$)

Reset value: 0x0003 0000

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

All the bits of HPDF_FLTySTAT are automatically reset when FLTN=0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
Reserved				MMF[1:0]		Reserved															CKLF[1:0]
r															r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reserved	RCPF	ICPF	Reserved				TMEOF				RCOF	ICOF	RCEF	ICEF							
r	r						r				r	r	r	r	r						

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value
25:24	MMF[1:0]	<p>Malfunction monitor flag</p> <p>00: No malfunction event occurred on channel 0 and channel 1</p> <p>01: Malfunction event occurred on channel 0</p> <p>10: Malfunction event occurred on channel 1</p> <p>11: Malfunction event occurred on channel 0 and channel 1</p> <p>This bit is set by hardware. It can be cleared by setting the MMFC[1:0] bit field in the HPDF_FLTyINTC register.</p> <p>This bit is also cleared by hardware when CHEN= 0.</p> <p>This bit is only available in HPDF_FTL0STAT register.</p>
23:18	Reserved	Must be kept at reset value
17:16	CKLF[1:0]	<p>Clock loss flag</p> <p>00: Clock signal is not lost on channel 0 and channel 1</p> <p>01: Clock signal is lost on channel 0</p> <p>10: Clock signal is lost on channel 1</p> <p>11: Clock signal is lost on channel 0 and channel 1</p> <p>When CHEN=0 or the serial interface is not synchronized, the state is maintained</p>

by the hardware. After the synchronization of serial interface is completed, if the clock of channel x is lost, the corresponding bit in CKLF [1:0] bit field are set by hardware. By setting the CKLFC[1:0] bit field in HPDF_FLTyINTC, the corresponding bit in the CKLF[1:0] bit field can be cleared.

This bit is only available in HPDF_FLT0STAT register.

15	Reserved	Must be kept at reset value
14	RCPF	<p>Regular conversion in progress flag</p> <p>0: No request of regular conversion has been generated</p> <p>1: The regular conversion is in progress or a request for a regular conversion is pending</p> <p>If RCPF=1, a request to start a regular conversion is ignored. When write 1 to SRCS bit, the RCPF will be setted 1 immediately.</p>
13	ICPF	<p>Inserted conversion in progress flag</p> <p>0: No request to the inserted group conversion has been generated (neither by software nor by trigger).</p> <p>1: The inserted group conversion is in progress or a request for a inserted conversion is pending.</p> <p>If ICPF=1, a request to start an inserted conversion is ignored. When write 1 to SICC bit, the ICPF will be setted 1 immediately.</p>
12:5	Reserved	Must be kept at reset value
4	TMEOF	<p>Threshold monitor event occurred flag</p> <p>0: No Threshold monitor event occurred</p> <p>1: Threshold monitor event occurred which detected data crosses the threshold</p> <p>This bit is set by hardware.</p> <p>It is cleared by clearing HTF[1:0] and LTF[1:0] in HPDF_FLTyTMSTAT register.</p>
3	RCDOF	<p>Regular conversion data overflow flag</p> <p>0: No regular conversion data overflow has occurred</p> <p>1: A regular conversion data overflow has occurred</p> <p>If RCDOF=1, it means that a regular conversion finished while RCEF has already been set. RDATA is not affected by overflows.</p> <p>This bit is set by hardware.</p> <p>It can be cleared by setting the RCDOFC in the HPDF_FLTyINTC register.</p>
2	ICDOF	<p>Inserted conversion data overflow flag</p> <p>0: No inserted conversion data overflow has occurred</p> <p>1: An inserted conversion data overflow has occurred</p> <p>If ICDOF=1, it means that an inserted conversion finished while ICEF has already been set. FLTyIDATA is not affected by overflows</p> <p>This bit is set by hardware.</p> <p>It can be cleared by software setting the ICDOFC bit in the HPDF_FLTyINTC register.</p>

1	RCEF	Regular conversion end flag 0: No regular conversion has completed 1: A regular conversion has completed If RCEF=1, it means that the data may be read. This bit is set by hardware. It is cleared when the software or DMA reads HPDF_FLTyRDATA register.
0	ICEF	Inserted conversion end flag 0: No inserted conversion has completed 1: An inserted conversion has completed If ICEF=1, it means that its data may be read. This bit is set by hardware. It is cleared when the software or DMA reads HPDF_FLTyIDATA register.

Filter y interrupt flag clear register (HPDF_FLTyINTC)

Address offset: 0x10C + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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

Note: The bits of HPDF_FLTyINTC are always read as '0'.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				MMFC[1:0]				Reserved				CKLFC[1:0]			
rc_w1															rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RCOFC		ICOFC		Reserved	
rc_w1															rc_w1

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	MMFC[1:0]	Clear the malfunction monitor flag 00: No effect 01: Clear the malfunction monitor flag on channel 0 10: Clear the malfunction monitor flag on channel 1 11: Clear the malfunction monitor flag on channel 0 and channel 1 This bit is only available in HPDF_FLT0INTC register.
23:18	Reserved	Must be kept at reset value.
17:16	CKLFC[1:0]	Clear the clock loss flag 00: No effect 01: Clear the clock loss flag on channel 0 10: Clear the clock loss flag on channel 1 11: Clear the clock loss flag on channel 0 and channel 1.

When the serial transceiver is not yet synchronized, the clock loss flag is set and cannot be cleared by CKLFC[1:0].

This bit is only available in HPDF_FLT0INTC register.

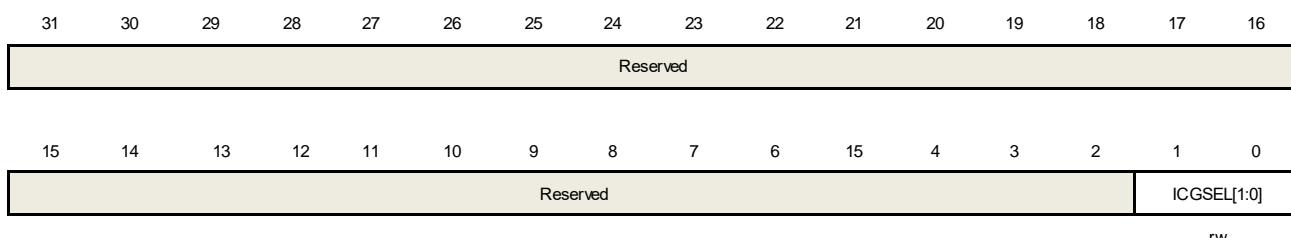
15:4	Reserved	Must be kept at reset value.
3	RCDOFC	Clear the regular conversion data overflow flag 0: No effect 1: Clear the RCDOF bit in the HPDF_FLTySTAT register
2	ICDOFC	Clear the inserted conversion data overflow flag 0: No effect 1: Clear the ICDOF bit in the HPDF_FLTySTAT register
1:0	Reserved	Must be kept at reset value.

Filter y inserted channel group selection register (HPDF_FLTyICGS)

Address offset: $0x110 + 0x80 * y$, ($y = 0, 1$)

Reset value: 0x0000 0001

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



rw

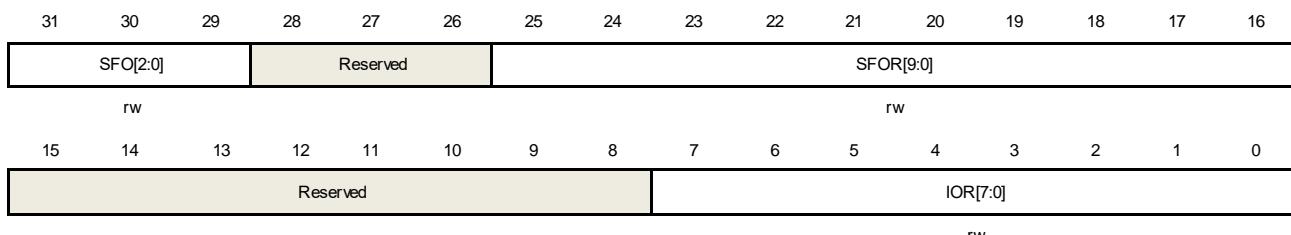
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	ICGSEL[1:0]	Inserted channel group selection 01: Channel 0 belongs to the inserted group 10: Channel 1 belongs to the inserted group 11: Channel 0 and channel 1 belong to the inserted group If SCMOD=1, each of the selected channels is converted, one after another. The priority conversion with lowest channel number. If SCMOD=0, then only one channel is converted from the selected channels, and the channel selection is moved to the next channel. When SCMOD=0, Writing ICGSEL will reset the channel selection to the lowest selected channel. At least one channel must always be selected for the inserted group. All writes that make ICGSEL[1:0]=0 are ignored.

Filter y sinc filter configuration register (HPDF_FLTySFCFG)

Address offset: 0x114 + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:29	SFO[2:0]	Sinc filter order 000: FastSinc filter type 001: Sinc ¹ filter type 010: Sinc ² filter type 011: Sinc ³ filter type 100: Sinc ⁴ filter type 101: Sinc ⁵ filter type 110~111: Reserved This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.
28:26	Reserved	Must be kept at reset value.
25:16	SFOR[9:0]	Sinc filter oversampling ratio (decimation rate) 0~1023: Sinc filter oversampling ratio (decimation rate) SFOR= SFOR[9:0] +1. If SFOR [9:0] = 0 (SFOR=1), the filter will be bypass. This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.
15:8	Reserved	Must be kept at reset value.
7:0	IOR[7:0]	Integrator oversampling ratio 0~255: Integrator oversampling ratio IOR=IOR[7:0]+1. The output data rate from the integrator will be decreased by this value. If IOR[7:0] = 0 (IOR=1), the integrator will be bypass. This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.

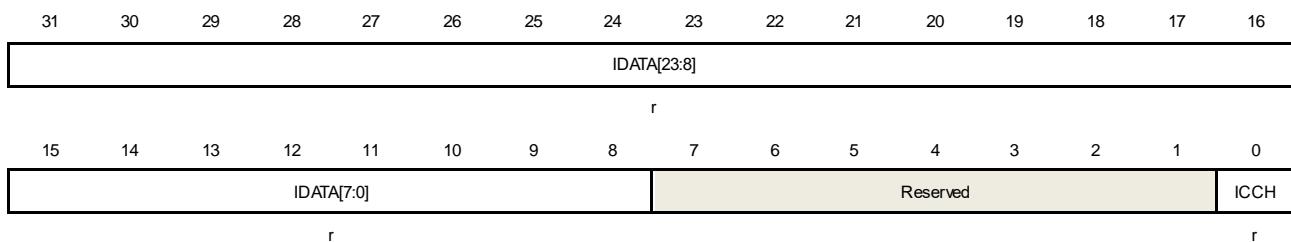
Filter y inserted group conversion data register (HPDF_FLTyIDATA)

Address offset: 0x118 + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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

Note: Half-word accesses may be used to read only the MSB of conversion data. DMA can be used to read the data from this register. Reading this register also clears ICEF bit.



Bits	Fields	Descriptions
31:8	IDATA[23:0]	Inserted group conversion data When each a channel in the inserted group is converted, the resulting data is stored in this field. The data is valid when ICEF=1. Reading this register clears the corresponding ICEF (in HPDF_FLTySTAT register).
7:1	Reserved	Must be kept at reset value.
0	ICCH	Inserted channel most recently converted When each a channel in the inserted group is converted, ICCH is updated to indicate which channel was converted. Therefore, IDATA[23:0] holds the data that corresponds to the channel indicated by ICCH.

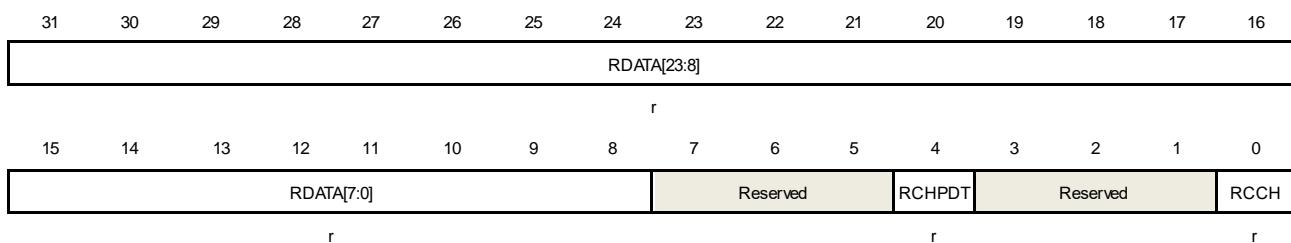
Filter y regular channel conversion data register (HPDF_FLTyRDATA)

Address offset: 0x11C + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

This register has to be accessed by half-word (16-bit). or word (32-bit).

Note: Half-word accesses may be used to read only the MSB of conversion data. Reading this register also clears RCEF bit.



Bits	Fields	Descriptions
31:8	RDATA[23:0]	Regular channel conversion data When each regular conversion finishes, its data is stored in these bits. The data is valid when RCEF=1. Reading this register clears the corresponding RCEF (in HPDF_FLTySTAT register).

7:5	Reserved	Must be kept at reset value.
4	RCHPDT	Regular channel pending data Regular data in RDATA[23:0] was delayed due to an inserted channel trigger during the conversion
3:1	Reserved	Must be kept at reset value.
0	RCCH	Regular channel most recently converted When each regular conversion finishes, RCCH is updated to indicate which channel was converted. Thus RDATA[23:0] holds the data that corresponds to the channel indicated by RCCH.

Filter y threshold monitor high threshold register (HPDF_FLTyTMHT)

Address offset: 0x120 + 0x80 * y, (y = 0, 1)

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
HTVAL[23:8]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTVAL[7:0]								Reserved				HTBSD[1:0]			
rw								rw							

Bits	Fields	Descriptions
31:8	HTVAL[23:0]	Threshold monitor high threshold value These bits are written by software to determine the high threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits HTVAL[7:0] are ignored.
7:2	Reserved	Must be kept at reset value.
1:0	HTBSD[1:0]	High threshold event break signal distribution 00: Break signal is not distributed to high threshold event 01: Break signal 0 is distributed to high threshold event 10: Break signal 1 is distributed to high threshold event 11: Break signal 0 and 1 are distributed to high threshold event

Filter y threshold monitor low threshold register (HPDF_FLTyTMLT)

Address offset: 0x124 + 0x80 * y, (y = 0, 1)

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
LVAL[23:8]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVAL[7:0]								Reserved				LTBSD[1:0]			
rw															

Bits	Fields	Descriptions
31:8	LVAL[23:0]	Threshold monitor low threshold value. These bits are written by software to determine the low threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits LVAL[7:0] are ignored.
7:2	Reserved	Must be kept at reset value.
1:0	LTBSD[1:0]	Low threshold event break signal distribution 00: Break signal is not distributed to low threshold event 01: Break signal 0 is distributed to low threshold event 10: Break signal 1 is distributed to low threshold event 11: Break signal 0 and 1 is distributed to low threshold event

Filter y threshold monitor status register (HPDF_FLTyTMSTAT)

Address offset: 0x128 + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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

Note: All the bits of HPDF_FLTyTMSTAT are automatically reset when FLTN=0.

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

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	HTF[1:0]	Threshold monitor high threshold flag 00: No high threshold error on channel 0 and channel 1 01: High threshold error on channel 0 10: High threshold error on channel 1

11: High threshold error on channel 0 and channel 1
 It is set by hardware. It can be cleared by software setting the corresponding HTFC[1:0] bit in the HPDF_FLTyTMFC register.

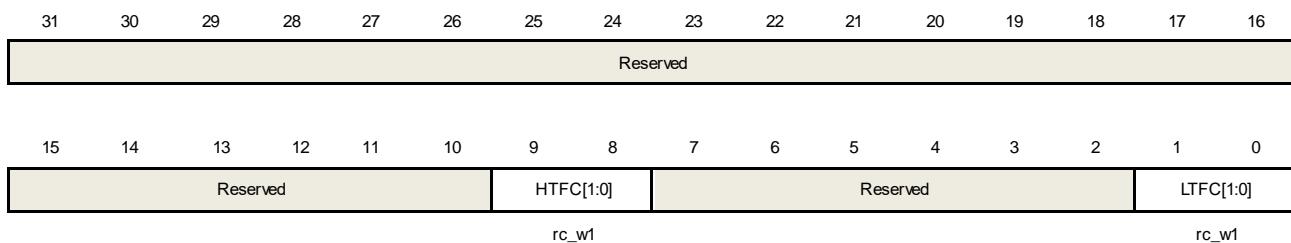
7:2	Reserved	Must be kept at reset value.
1:0	LTF[1:0]	<p>Threshold monitor low threshold flag</p> <p>00: No low threshold error on channel 0 and channel 1</p> <p>01: Low threshold error on channel 0</p> <p>10: Low threshold error on channel 1</p> <p>11: Low threshold error on channel 0 and channel 1</p> <p>It is set by hardware. It can be cleared by software setting the corresponding LTFC[1:0] bit in the HPDF_FLTyTMFC register.</p>

Filter y threshold monitor flag clear register (HPDF_FLTyTMFC)

Address offset: 0x12C + 0x80 * y, (y = 0, 1)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	HTFC[1:0]	<p>Clear the threshold monitor high threshold flag</p> <p>00: No effect</p> <p>01: Clear the threshold monitor high threshold flag on channel 0</p> <p>10: Clear the threshold monitor high threshold flag on channel 1</p> <p>11: Clear the threshold monitor high threshold flag on channel 0 and channel 1</p>
7:2	Reserved	Must be kept at reset value.
1:0	LTF[1:0]	<p>Clear the threshold monitor low threshold flag</p> <p>00: No effect</p> <p>01: Clear the threshold monitor low threshold flag on channel 0</p> <p>10: Clear the threshold monitor low threshold flag on channel 1</p> <p>11: Clear the threshold monitor low threshold flag on channel 0 and channel 1</p>

Filter y extremes monitor maximum register (HPDF_FLTyEMMAX)

Address offset: $0x130 + 0x80 * y$, ($y = 0, 1$)

Reset value: 0x8000 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
MAXVAL[23:8]															
rc_r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAXVAL[7:0]								Reserved							
rc_r															

Bits	Fields	Descriptions
31:8	MAXVAL[23:0]	<p>Extremes monitor maximum value</p> <p>These bits are set by hardware and indicate the highest value of channel converted by HPDF_FLTy.</p> <p>These bits can be reset by reading of this register.</p>
7:1	Reserved	Must be kept at reset value.
0	MAXDC	<p>Extremes monitor maximum data channel.</p> <p>This bits indicate the channel on which the data is stored into MAXVAL[23:0]. It can be cleared by reading of this register.</p>

Filter y extremes monitor minimum register (HPDF_FLTyEMMIN)

Address offset: $0x134 + 0x80 * y$, ($y = 0, 1$)

Reset value: 0x7FFF FF00

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
MINVAL[23:8]															
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MINVAL[7:0]								Reserved							
rs															

Bits	Fields	Descriptions
31:8	MINVAL[23:0]	<p>Extremes monitor minimum value</p> <p>These bits are set by hardware and indicate the lowest value converted by HPDF_FLTy.</p> <p>These bits can be reset by reading of this register.</p>

7:1	Reserved	Must be kept at reset value.
0	MINDC	<p>Extremes monitor minimum data channel</p> <p>This bit indicate the channel on which the data is stored into MINVAL[23:0]. It can be cleared by reading of this register.</p>

31. Infrared ray port (IFRP)

31.1. Overview

Infrared ray port (IFRP) is used to control infrared light LED, and send out infrared data to implement infrared ray remote control.

There is no register in this module, which is controlled by TIMER15 and TIMER16. The IFRP_OUT pin can be configured by GPIO alternate function selected register.

31.2. Characteristics

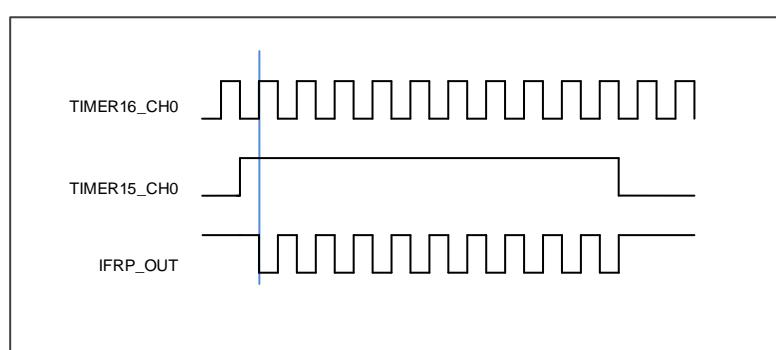
- The IFRP output signal is decided by TIMER15_CH0 and TIMER16_CH0
- To get correct infrared ray signal, TIMER15 should generate low frequency modulation envelope signal, and TIMER16 should generate high frequency carrier signal

31.3. Function overview

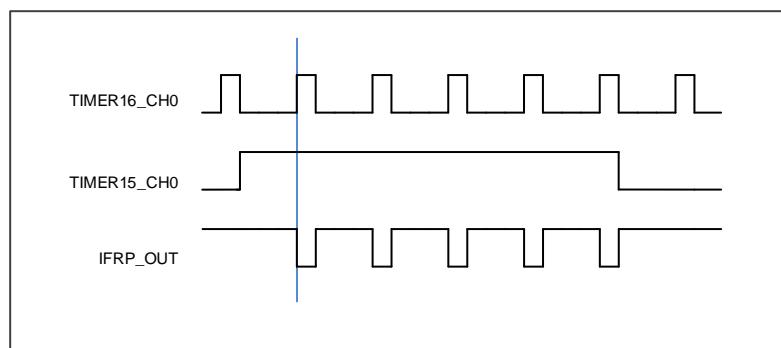
IFRP is a module which is able to integrate the output of TIMER15 and TIMER16 to generate an infrared ray signal.

1. The TIMER15's CH0 is programmed to generate the low frequency PWM signal which is the modulation envelope signal. The TIMER16's CH0 is programmed to generate the high frequency PWM signal which is the carrier signal. And the channel need to be enabled before generating these signals.
2. Program the GPIO alternate function selected register and enable the IFRP_OUT pin.

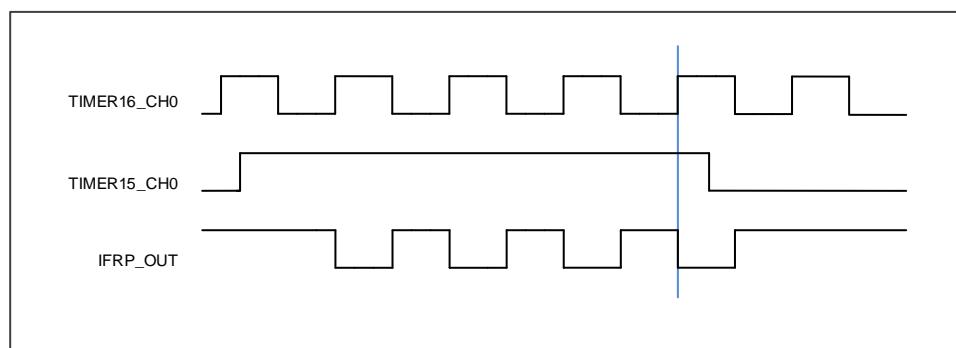
Figure 31-1. IFRP output timechart 1



Note: IFRP_OUT has one APB clock delay from TIMER16_CH0.

Figure 31-2. IFRP output timechart 2

Note: Carrier (TIMER15_CH0)'s duty cycle can be changed, and IFRP_OUT has inverted relationship with TIMER16_CH0 when TIMER15_CH0 is high.

Figure 31-3. IFRP output timechart 3

Note: IFRP_OUT will keep the integrity of TIMER16_CH0, even if envelope signal (TIMER15_CH0) is no active.

32. Wi-Fi

32.1. Overview

The GD32W51x is a highly integrated 2.4GHz Wi-Fi System-on-Chip (SoC) that includes an Arm® Cortex®-M33 processor with Trustzone, a single stream IEEE 802.11b/g/n MAC/baseband/radio, a power amplifier (PA), and a receive low-noise amplifier (LNA). It is an optimized SoC designed for a broad array of smart devices for Internet of Things (IoT) applications.

32.2. Characteristics

32.2.1. Standards Supported

- 802.11b/g/n(2.4G) compatible
- 802.11e QoS Enhancement (WMM)
- 802.11i (WPA, WPA2). Open, shared key, and pair-wise key authentication services
- Wi-Fi WPS
- Wi-Fi Direct
- Integrated TCP/IP protocol

32.2.2. Wi-Fi MAC

- Transmission and reception of aggregated MPDUs (A-MPDU) for high throughput (HT)
- Support for immediate ACK and Block-ACK policies
- Support for power management schemes, including WMM power-save, power-save multi-poll (PSMP), and multiphase PSMP operation
- Interframe space timing support, including RIFS
- Support for RTS/CTS and CTS-to-self frame sequences for protecting frame exchanges
- Back-off counters in hardware for supporting multiple priorities as specified in the WMM specification
- Timing synchronization function (TSF), network allocation vector (NAV) maintenance, and target beacon transmission time (TBTT) generation in hardware
- Hardware engine for AES-CCMP, legacy WPA TKIP, legacy WEP ciphers, and support for key management
- Programmable independent basic service set (IBSS) or infrastructure basic service set or Access Point functionality

32.2.3. Wi-Fi PHY

- Single antenna 1x1 stream in 20MHz and 40MHz channels
- Supports IEEE 802.11b DSSS-CCK modulation: 1, 2, 5.5, 11Mbps

- Supports IEEE 802.11g OFDM modulation: 6, 9, 12, 18, 24, 36, 48, 54Mbps
- Supports IEEE 802.11n HT modulations MCS0-7, 20MHz, 800ns guard interval: 6.5, 13.0, 19.5, 26, 39, 52.0, 58.5, 65.0Mbps
- Supports IEEE 802.11n HT modulations MCS0-7, 20MHz, 400ns guard interval: 7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2Mbps
- Supports IEEE 802.11n HT modulations MCS0-7, 40MHz, 800ns guard interval: 13.5, 27, 40.5, 54, 81, 108, 121.5, 135Mbps
- Supports IEEE 802.11n HT modulations MCS0-7, 40MHz, 400ns guard interval: 15, 30, 45, 60, 90, 120, 135, 150Mbps
- IEEE 802.11n mixed mode operation
- Per packet TX power control
- Advanced channel estimation/equalization, automatic gain control, CCA, carrier/symbol recovery, and frame detection
- Digital calibration algorithms to handle CMOS RF chip process, voltage, and temperature (PVT) variations
- Per-packet channel quality and signal-strength measurements
- Compliance with FCC and other worldwide regulatory requirements

32.2.4. Wi-Fi Radio

- Fractional-N for multiple reference clock support
- Integrated PA with power control
- Optimized Tx gain distribution for linearity and noise performance
- Direct conversion architecture
- On-chip gain selectable LNA with optimized noise figure
- High dynamic range AGC
- Frequency Range 2.4G-2.5G

33. Document appendix

33.1. List of abbreviations used in registers

Table 33-1. 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_w 1)	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_w 0)	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.

33.2. List of terms

Table 33-2. 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.
RAZ/WI	Read-as-zero, writes ignored.

33.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

34. Revision history

Table 34-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.25, 2021

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 under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. 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.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do 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 Products. Customers shall indemnify and hold the Company 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 Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.

© 2021 GigaDevice – All rights reserved