A31G324, A31G323
User's Manual

32-bit Cortex-M0+ based
General Purpose Microcontroller

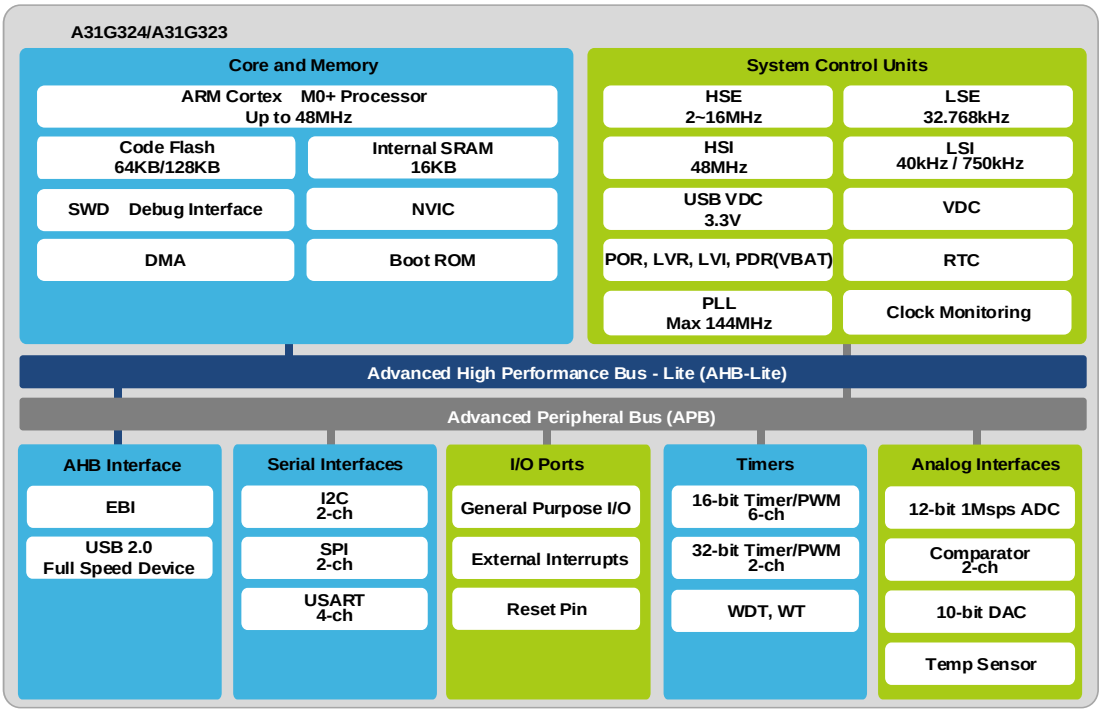
Version 1.08

Introduction

This user's manual contains complete information for application developers who use A31G324 or A31G323 for their specific needs.

A31G32x devices are 32-bit general purpose microcontrollers for various USB appliances. To meet the requirements for the complexity and high performance in consumer and portable electronics, A31G32x incorporate ARM's high-speed 32-bit Cortex-M0+ Cores, USB full speed interface, high speed ADC, various power mode for power consumption reduction, plentiful analog functions, and high resolution PWM for precision control and data processing.

Building on the very successful Cortex-M0+ processor, the ARM® Cortex®-M0+ retains full instruction set and tool compatibility, while further reducing energy consumption and increasing performance.



Reference document

- Document 'DDI 0484C' is provided by ARM and contains information of Cortex-M0+.

Contents

Introduction 1
Reference document 1
1 Description 19
1.1 Device overview 19
1.2 Block diagram 22
2 Pinouts and pin descriptions 23
2.1 Pinouts 23
2.1.1 A31G323RLN, A31G324RLN (64 LQFP) 23
2.1.2 A31G323CLN, A31G324CLN (48 LQFP) 24
2.1.3 A31G323CUN, A31G324CUN (48 QFN) 25
2.2 Pin description 26
3 System and memory overview 35
3.1 System architecture 35
3.1.1 Cortex-M0+ Core 35
3.1.2 Interrupt controller 35
3.2 Memory organization 38
3.2.1 Register boundary address 38
3.2.2 Memory map 39
3.2.3 Embedded SRAM 40
3.2.4 Flash memory overview 40
3.2.5 Boot mode 40
4 System control unit 43
4.1 SCU block diagram 43
4.2 Clock system 44
4.2.1 HCLK clock domain 45
4.2.2 PCLK clock domain 45
4.2.3 Clock configuration procedure 45
4.3 Power domain 48
4.4 Reset 49
4.4.1 Cold reset 50
4.4.2 Warm reset 51
4.4.3 LVR reset 51
4.4.4 Reset tree 52
4.5 Operation mode 53
4.5.1 RUN mode 54
4.5.2 SLEEP mode 54
4.5.3 Power-down mode 55
4.5.4 STOP mode 57
4.5.5 STANDBY mode 59
4.5.6 BACKUP POWER mode 60
4.6 Registers 62
4.6.1 SCUCC_VENDORID: vendor ID register 65
4.6.2 SCUCC_CHIPID: chip ID register 65
4.6.3 SCUCC_REVNR: revision number register 66
4.6.4 SCU_SMR: system mode register 66
4.6.5 SCU_SCR: system control register 67
4.6.6 SCU_WUER: wakeup source enable register 67
4.6.7 SCU_WUSR: wakeup source status register 69

4.6.8 SCU_RSER: reset source enable register 71
4.6.9 SCU_RSSR: reset source status register 72
4.6.10 SCU_PRER1: peripheral reset enable register 1 73
4.6.11 SCU_PRER2: peripheral reset enable register 2 74
4.6.12 SCU_PER1: peripheral enable register1 75
4.6.13 SCU_PER2: peripheral enable register 2 76
4.6.14 SCU_PCER1: peripheral clock enable register 1 77
4.6.15 SCU_PCER2: peripheral clock register 2 78
4.6.16 SCU_PPCLKSR: peripheral clock selection register 79
4.6.17 SCU_CSCR: clock source control register 80
4.6.18 SCU_SCCR: system clock control register 81
4.6.19 SCU_CMR: clock monitoring register 81
4.6.20 SCU_NMIR: NMI control register 83
4.6.21 SCU_COR: clock output register 84
4.6.22 SCU_PLLCON: PLL control register 85
4.6.23 SCU_VDCCON: VDC control register 87
4.6.24 SCU_PDRCON: PDR control register 88
4.6.25 SCU_LSICON: Low speed internal OSC control register 89
4.6.26 SCU_EOSCR: External oscillator control register 89
4.6.27 SCU_EMODR: External mode status register 90
4.6.28 SCU_RSTDBCR: pin reset debounce control register 91
4.6.29 SCU_MCCR1: Miscellaneous clock control register 1 92
4.6.30 SCU_MCCR2: Miscellaneous clock control register 2 93
4.6.31 SCU_MCCR3: Miscellaneous clock control register 3 94
4.6.32 SCU_MCCR4: Miscellaneous clock control register 4 95
4.6.33 SCU_MCCR5: Miscellaneous clock control register 5 96
4.6.34 SCU_MCCR6: Miscellaneous clock control register 6 97
4.6.35 SCULV_LVICR: Low voltage indicator control register 98
4.6.36 SCULV_LVRCR: Low voltage reset control register 99
4.6.37 SCULV_LVRCNFIG: Configuration for low voltage reset 100
5 PCU and GPIO 101
5.1 PCU and GPIO Block diagram 101
5.2 Pin multiplexing 103
5.3 Registers 105
5.3.1 Pn_MOD: PORT n mode register 106
5.3.2 Pn_TYP: Port n output type selection register 106
5.3.3 Pn_AFSR1: Port n alternative function selection register 1 107
5.3.4 Pn_AFSR2: Port n alternative function selection register 2 108
5.3.5 Pn_PUPD: Port n pull-up/down resistor selection register 109
5.3.6 Pn_INDR: Port n input data register 109
5.3.7 Pn_OUTDR: Port n output data register 110
5.3.8 Pn_BSR: Port n output bit set register 110
5.3.9 Pn_BCR: Port n output bit clear register 111
5.3.10 Pn_OUTDMSK: Port n output data mask register 111
5.3.11 Pn_DBCR: Port n debounce control register 112
5.3.12 Pn_IER: Port n interrupt enable register 112
5.3.13 Pn_ISR: Port n interrupt status register 113
5.3.14 Pn_ICR: Port n interrupt control register 113
5.3.15 PCU_USBCON: USB control register 114
5.3.16 PCU_PORTEN: Port access enable 115

5.4 Functional description ... 116
6 Flash memory controller ... 118
6.1 Registers ... 120
6.1.1 FMC_MR: Flash memory mode register ... 121
6.1.2 FMC_CR: Flash memory control register ... 121
6.1.3 FMC_AR: Flash memory address register ... 122
6.1.4 FMC_DR: Flash data input register ... 122
6.1.5 FMC_BUSY: Flash write busy status register ... 123
6.1.6 FMC_CRC: Flash CRC check register ... 123
6.1.7 FMC_CFG: Flash memory configuration register ... 124
6.1.8 FMC_WPROT: Write protection register ... 124
6.1.9 FMC_LOCK: Flash lock register ... 125
6.2 Functional description ... 126
6.2.1 Flash erase and program examples ... 126
7 Direct Memory Access Controller (DMAC) ... 129
7.1 Block diagram ... 129
7.2 Registers ... 130
7.2.1 DCn.CR: DMA controller configuration register ... 130
7.2.2 DCn.SR: DMA controller status register ... 132
7.2.3 DCn.PAR: DMA controller peripheral address register ... 132
7.2.4 DCn.MAR: DMA controller memory address register ... 133
7.3 Functional description ... 134
7.3.1 DMA operation ... 135
8 Static memory controller (EBI) ... 138
8.1 Block diagram ... 139
8.2 Registers ... 139
8.2.1 SMIBCRn: Memory area configuration register n ... 140
8.3 Functional description ... 141
8.3.1 Chip select signal ... 141
8.3.2 Byte lane control and control signal ... 141
8.3.3 Bus control signal in read operation ... 144
8.3.4 Bus control signal in write operation ... 147
8.3.5 Timing change of control signals ... 150
8.3.6 Wait control while accessing memory ... 151
9 Watchdog timer (WDT) ... 152
9.1 WDT block diagram ... 152
9.2 Registers ... 153
9.2.1 WDT_CR: Watchdog timer control register ... 154
9.2.2 WDT_SR: Watchdog timer status register ... 155
9.2.3 WDT_DR: Watchdog timer data register ... 155
9.2.4 WDT_CNT: Watchdog timer counter register ... 156
9.2.5 WDT_WINDR: Watchdog timer window data register ... 156
9.2.6 WDT_CNTR: Watchdog timer counter reload register ... 157
9.3 Functional description ... 157
9.3.1 Timing diagram ... 158
9.3.2 Prescale table ... 158
10 Watch timer ... 159
10.1 WT block diagram ... 159
10.2 Registers ... 160
10.2.1 WT_CR: Watch timer control register ... 161

10.2.2 WT_DR: Watch timer data register 162
10.2.3 WT_CNT: Watch timer counter register 162
11 16-bit timer 163
11.1 16-bit timer block diagram 164
11.2 Registers 164
11.2.1 TIMER1n_CR: Timer/counter 1n control register 165
11.2.2 TIMER1n_ADR: Timer/counter 1n A data register 167
11.2.3 TIMER1n_BDR: Timer/counter n B data register 167
11.2.4 TIMER1n_CAPDR: Timer/counter 1n capture data register 168
11.2.5 TIMER1n_PREDR: Timer/counter 1n prescaler data register 168
11.2.6 TIMER1n_CNT: Timer/counter n counter register 168
11.3 Functional description 169
11.3.1 Timer counter 10/11/12/13 169
11.3.2 16-bit timer/counter mode 170
11.3.3 16-bit capture mode 172
11.3.4 16-bit PPG mode 175
12 32-bit timer 178
12.1 32-bit timer block diagram 179
12.2 Registers 180
12.2.1 TIMER2n_CR: Timer/counter 2n control register 181
12.2.2 TIMER2n_ADR: Timer/counter 2n A data register 182
12.2.3 TIMER2n_BDR: Timer/counter 2n B data register 183
12.2.4 TIMER2n_CAPDR: Timer/counter 2n capture data register 183
12.2.5 TIMER2n_PREDR: Timer/counter 2n prescaler data register 184
12.2.6 TIMER2n_CNT: Timer/counter 2n counter register 184
12.3 Functional description 185
12.3.1 Timer counter 2n 185
12.3.2 32-bit timer/counter mode 186
12.3.3 32-bit capture mode 188
12.3.4 32-bit PPG mode 191
13 Timer counter 30 194
13.1 Timer counter 30 block diagram 195
13.2 Registers 196
13.2.1 TIMER30_CR: Timer/counter 30 control register 197
13.2.2 TIMER30_PDR: Timer/counter 30 period data register 199
13.2.3 TIMER30_ADR: Timer/counter 30 A data register 199
13.2.4 TIMER30_BDR: Timer/counter 30 B data register 199
13.2.5 TIMER30_CDR: Timer/counter 30 C data register 200
13.2.6 TIMER30_CAPDR: Timer/counter 30 capture data register 200
13.2.7 TIMER30_PREDR: Timer/counter 30 prescaler data register 200
13.2.8 TIMER30_CNT: Timer/counter 30 counter register 201
13.2.9 TIMER30_OUTCR: Timer/counter 30 output control register 201
13.2.10 TIMER30_DLY: Timer/counter 30 PWM output delay data register 202
13.2.11 TIMER30_INTCR: Timer/counter 30 interrupt control register 203
13.2.12 TIMER30_INTFLAG: Timer/counter 30 interrupt flag register 204
13.2.13 TIMER30_HIZCR: Timer/counter 30 high-impedance control register 205
13.2.14 TIMER30_ADTCR: Timer/counter 30 A/DC trigger control register 206
13.2.15 TIMER30_ADTDR: Timer/counter 30 A/DC trigger generator data register 207
13.3 Functional description 208
13.3.1 Timer counter 30 208

13.3.2 Timer 30 capture mode 208
13.3.3 Timer 30 interval mode 211
14 Timer counter 40 223
14.1 Timer counter 40 block diagram 224
14.2 Registers 225
14.2.1 TIMER40_CR: Timer/counter 40 control register 226
14.2.2 TIMER40_ADR: Timer/counter 40 A data register 227
14.2.3 TIMER40_BDR: Timer/counter 40 B data register 228
14.2.4 TIMER40_CAPDR: Timer/counter 40 capture data register 228
14.2.5 TIMER40_PREDR: Timer/counter 40 prescaler data register 228
14.2.6 TIMER40_PREDR2: Timer/counter 40 prescaler data register 2 229
14.2.7 TIMER40_CNT: Timer/counter 40 counter register 229
14.3 Functional description 230
14.3.1 Timer counter 40 230
14.3.2 16-bit timer/counter mode 230
14.3.3 16-bit capture mode 233
14.3.4 16-bit PPG mode 236
15 Universal synchronous/asynchronous receiver/transmitter 239
15.1 USART block diagram 240
15.2 Registers 241
15.2.1 USARTn_CR1: USARTn control register 1 242
15.2.2 USARTn_CR2: USARTn control register 2 244
15.2.3 USARTn_ST: USARTn status register 245
15.2.4 USARTn_BDR: USARTn baud rate generation register 246
15.2.5 USARTn_DR: USARTn data register 247
15.2.6 USARTn_FPCR: USARTn floating point count register 247
15.2.7 USARTn_RTO: USARTn RTO register 248
15.3 Functional description 248
15.3.1 USART clock generation 249
15.3.2 External clock (SCKn) 250
15.3.3 Synchronous mode operation 250
15.3.4 UART data format 251
15.3.5 UART parity bit 252
15.3.6 UART transmitter 252
15.3.7 UART receiver 254
15.3.8 SPI mode 257
15.3.9 SPI clock formats and timing 257
15.3.10 Baud rate settings (example) 260
15.3.11 0% error baud rate 261
15.3.12 Receive time out (RTO) 261
16 I2C interface 262
16.1 I2C block diagram 263
16.2 Registers 263
16.2.1 I2Cn_CR: I2Cn control register 264
16.2.2 I2Cn_ST: I2Cn status register 265
16.2.3 I2Cn_SAR1: I2Cn slave address register 1 266
16.2.4 I2Cn_SAR2: I2Cn slave address register 2 267
16.2.5 I2Cn_DR: I2Cn data register 267
16.2.6 I2Cn_SDAHR: I2Cn SDA hold time register 268
16.2.7 I2Cn_SCLLR: I2Cn SCL low period register 268

16.2.8 I2Cn_SCLHR: I2Cn SCL high period register 269
16.2.9 I2Cn_MR: I2Cn mode control register 269
16.3 Functional description 270
16.3.1 I2C bit transfer 270
16.3.2 START/repeated START/STOP 270
16.3.3 Data transfer 271
16.3.4 Acknowledge 271
16.3.5 Synchronization/Arbitration 272
16.3.6 I2C operation 273
17 Serial peripheral interface (SPI) 280
17.1 SPI block diagram 281
17.2 Registers 281
17.2.1 SPIn.TDR: SPI transmit data register 282
17.2.2 SPIn.RDR: SPI receive data register 282
17.2.3 SPIn.CR: SPI control register 282
17.2.4 SPIn.SR: SPI status register 284
17.2.5 SPIn.BR: SPI baud rate register 285
17.2.6 SPIn.EN: SPI enable register 286
17.2.7 SPIn.LR: SPI delay length register 286
17.3 Functional description 287
17.3.1 SPI timing 287
17.3.2 DMA handshake 290
18 12-bit ADC 291
18.1 12-bit ADC block diagram 292
18.2 Registers 292
18.2.1 ADC_MR: ADC mode register 294
18.2.2 ADC_CSCR: ADC current sequence/channel register 295
18.2.3 ADC_CCR: ADC clock control register 296
18.2.4 ADC_TRG: ADC trigger selection register 297
18.2.5 ADC_SCSR1: ADC channel selection 1 register 298
18.2.6 ADC_SCSR2: ADC channel selection 2 register 299
18.2.7 ADC_CR: ADC control register 299
18.2.8 ADC_SR: ADC state register 300
18.2.9 ADC_IER: ADC interrupt enable register 301
18.2.10 ADC_DDR: ADC DMA data register 301
18.2.11 ADC_DR: ADC sequence 0 to 7 data register 302
18.2.12 ADC_CMPR: ADC channel compare register 303
18.3 Functional description 304
18.3.1 ADC single mode timing diagram 304
18.3.2 ADC burst mode timing diagram 304
18.3.3 ADC sequential mode timing diagram 305
18.3.4 ADC multiple mode timing diagram 307
19 10-bit DAC 308
19.1 10-bit DAC block diagram and analog power pin 308
19.2 Registers 309
19.2.1 DAC_DR: DAC data register 309
19.2.2 DAC_BR: DAC buffer register 310
19.2.3 DAC_CR: DAC control register 310
19.2.4 DAC_PGSR: Programmable gain control register 312
19.2.5 DAC_OFSCR: DAC offset control register 312

19.3 Functional description ... 313
19.3.1 DAC data buffer register ... 313
19.3.2 Automatic DAC data increment/decrement ... 313
19.3.3 Programmable gain controller ... 314
20 Comparator ... 316
20.1 Comparator block diagram ... 316
20.2 Registers ... 317
20.2.1 COMP0CR: Comparator 0 control register ... 318
20.2.2 COMP1CR: Comparator 1 control register ... 319
20.2.3 COMPDBNC: Comparator debounce register ... 320
20.2.4 COMPICON: Comparator Interrupt control register ... 320
20.2.5 COMPIEN: Comparator interrupt enable register ... 321
20.2.6 COMPIST: Comparator interrupt status register ... 322
20.2.7 COMPICLR: Comparator interrupt clear register ... 322
21 Cyclic redundancy check and checksum (CRC checksum) ... 323
21.1 CRC and checksum block diagram ... 323
21.2 Registers ... 324
21.2.1 CRC_CR: CRC control register ... 324
21.2.2 CRC_IN: CRC input data register ... 325
21.2.3 CRC_RLT: CRC result data register ... 326
21.2.4 CRC_INIT: CRC initial data register ... 326
21.3 Functional description ... 327
21.3.1 CRC polynomial structure ... 327
21.3.2 CRC operation procedure in DMA mode ... 327
21.3.3 CRC operation procedure in user CRC and checksum mode ... 327
22 USB full speed device interface ... 328
22.1 USB block diagram ... 328
22.2 Registers ... 329
22.2.1 GOTGCTL: Control and status register ... 332
22.2.2 GOTGINT: Interrupt register ... 332
22.2.3 GAHBCFG: AHB configuration register ... 333
22.2.4 GUSBCFG: USB configuration register ... 335
22.2.5 GRSTCTL: Reset register ... 336
22.2.6 GINTSTS: Interrupt register ... 339
22.2.7 GINTMSK: Interrupt mask register ... 342
22.2.8 GRXSTSR: Receive status debug read register ... 345
22.2.9 GRXSTSP: Receive status read/pop register ... 346
22.2.10 GRXFSIZ: Receive FIFO size register ... 347
22.2.11 GNPTXFSIZ: Non-periodic transmit FIFO size register ... 348
22.2.12 GHWCFG1: User HW config1 register ... 348
22.2.13 GHWCFG2: User HW config2 register ... 349
22.2.14 GHWCFG3: User HW config2 register ... 349
22.2.15 GHWCFG4: User hardware config4 register ... 351
22.2.16 GLPMCFG: LPM config register ... 352
22.2.17 GPWRDN: Global power down register ... 356
22.2.18 GDFIFOCFG: Global DFIFO configuration register ... 360
22.2.19 DIEPTXFi: Device IN endpoint transmit FIFO size register i ... 360
22.2.20 DCFG: Device configuration register ... 361
22.2.21 DCTL: Device control register ... 364
22.2.22 DSTS: Device status register ... 368

22.2.23 DIEPMSK: Device IN endpoint common interrupt mask register 369
22.2.24 DOEPMSK: Device OUT endpoint common interrupt mask register 370
22.2.25 DAINT: Device all endpoints interrupt register 372
22.2.26 DAINTMSK: Device all endpoints interrupt mask register 373
22.2.27 DTHRCTL: Device threshold control register 374
22.2.28 DIEPEMPMSK: Device IN endpoint FIFO empty interrupt mask register 376
22.2.29 DIEPCTL0: Device control IN endpoint 0 control register 376
22.2.30 DIEPINT0: Device IN endpoint 0 interrupt register 378
22.2.31 DIEPDMA0: Device IN endpoint 0 DMA address register 380
22.2.32 DTXFSTS0: Device IN endpoint transmit FIFO status register 0 381
22.2.33 DIEPDMAB0: Device IN endpoint 16 buffer address register 381
22.2.34 DIEPCTLi : Device control IN endpoint i control register 382
22.2.35 DIEPINTi: Device IN endpoint I interrupt register 385
22.2.36 DIEPDMAi: Device IN endpoint I DMA address register 387
22.2.37 DTXFSTSi: Device IN endpoint transmit FIFO status register i 388
22.2.38 DIEPDMABi: Device IN endpoint I buffer address register 388
22.2.39 DOEPCTL0: Device control OUT endpoint 0 control register 389
22.2.40 DOEPINT0: Device OUT endpoint 0 interrupt register 391
22.2.41 DOEPDMA0: Device OUT endpoint 0 DMA address register 393
22.2.42 DOEPDMAB0: Device OUT Endpoint 16 Buffer Address Register 393
22.2.43 DOEPCTLi: Device control OUT endpoint I control register 394
22.2.44 DOEPINTi: Device OUT endpoint I interrupt register 397
22.2.45 DOEPDMAi: Device OUT Endpoint I DMA Address Register 399
22.2.46 DOEPDMABi: Device OUT Endpoint I Buffer Address Register 400
22.2.47 PCGCCTL: Power and Clock Gating Control Register 400
22.3 Functional description 403
22.3.1 USB device 403
22.3.2 USB descriptors 405
22.3.3 USB operation 406
22.3.4 USB transfer 408
22.3.5 USB software example 409
23 Real-time clock (RTC) 413
23.1 RTC block diagram 414
23.2 Registers 415
23.2.1 RTCCR: RTC control register 416
23.2.2 ERRCOR: Error correction register 418
23.2.3 SEC: Second counter register 419
23.2.4 MIN: Minute counter register 419
23.2.5 HOUR: Hour counter register 420
23.2.6 DAY: Day counter register 421
23.2.7 WEEK: Week counter register 422
23.2.8 MONTH: Month counter register 422
23.2.9 YEAR: Year counter register 423
23.2.10 ALARMW: Alarm register 423
23.2.11 SUBDR: Sub-second counter compare register 424
23.2.12 TIME_TS: TimeStamp time register 424
23.2.13 DATE_TS: TimeStamp date register 425
23.2.14 BKUPDR: Backup register 1 to 4 425
23.2.15 RTCPFCR: RTC pin function configuration register 426
23.3 Functional description 427

23.3.1 Starting the RTC Operation 427
23.3.2 RTC RW mode (read and write mode) 428
23.3.3 RTC alarm setting 431
23.3.4 RTC Constant-period matching 432
23.3.5 RTC Timestamp 432
23.3.6 RTC 1Hz output 433
23.3.7 Error correction of RTC 433
23.3.8 BKUPDR and RTCPFCR registers 435
24 Temp sensor 437
24.1 Temp sensor block diagram 437
24.2 Registers 438
24.2.1 TSENSECON: Temp sensor control register 438
24.2.2 TSREFPEROID: Temp sensor reference clock period register 439
24.2.3 TSSENSECNT: Temp sensor sense clock count register 439
24.3 Functional discription 440
24.3.1 How to calculate and convert to temperature 440
25 Electrical characteristics 441
25.1 Absolute maximum ratings 441
25.2 Recommended operating conditions 442
25.3 ADC characteristics 442
25.4 DAC characteristics 443
25.5 Power on reset characteristics 443
25.6 Low voltage reset/indicator characteristics 444
25.7 High frequency internal RC oscillator characteristics 445
25.8 Low frequency internal RC oscillator characteristics 445
25.9 DC electrical characteristics 446
25.10 Supply current characteristics 447
25.11 AC characteristics 448
25.12 USART SPI characteristics 449
25.13 I2C characteristics 451
25.14 USART UART timing characteristics 452
25.15 Data retention voltage in STOP mode 453
25.16 Internal Flash ROM characteristics 453
25.17 Main oscillator characteristics 454
25.18 Sub oscillator characteristics 456
25.19 Operating voltage range 457
25.20 PLL electrical characteristics 457
25.21 Comparator characteristics 458
25.22 USB characteristics 460
25.23 Temperature Sensor characteristics 460
26 Development tools 461
26.1 Compiler 461
26.2 Debugger 461
26.3 Programmer 462
27 Circuit Design Guide 463
28 Package information 464
28.1 64 LQFP package information 464
28.2 48 LQFP package information 466
28.3 48 QFN package information 468
29 Ordering information 470

Revision history 471

List of figures

Figure 1. A31G32x Block diagram 22
Figure 2. LQFP 64 Pinouts 23
Figure 3. LQFP 48 Pinouts 24
Figure 4. QFN 48 Pinouts 25
Figure 5. Interrupt Block Diagram 37
Figure 6. Memory Map 39
Figure 7. Connection Diagram of UART10 Boot 41
Figure 8. Connection Diagram of UART11 Boot 41
Figure 9. Connection Diagram of I2C Boot 42
Figure 10. Connection Diagram of E-PGM+ and SWD Port 42
Figure 11. SCU Block diagram 43
Figure 12. Clock Tree Configuration 44
Figure 13. Clock Change Procedure 46
Figure 14. Peripheral Clock Select (n: 1, 2, 3, 4, 5, and 6) 47
Figure 15. Power Domain Diagram 49
Figure 16. Power up Procedure 50
Figure 17. Warm Reset Diagram 51
Figure 18. LVR Reset Timing Diagram 51
Figure 19. Reset Tree Configuration 52
Figure 20. Transition between Operation Modes 53
Figure 21. SLEEP Mode Operation Sequence 54
Figure 22. Power down Mode Block Diagram 55
Figure 23. STOP Mode Sequence 58
Figure 24. STOP Mode Timing Diagram 58
Figure 25. STANDBY Mode Sequence 59
Figure 26. STANDBY Mode Timing Diagram 60
Figure 27. BACKUP POWER Mode Sequence 61
Figure 28. BACKUP POWER Mode Timing Diagram 62
Figure 29. LVI Block Diagram 99
Figure 30. LVR Block Diagram 99
Figure 31. PCU Block Diagram 101
Figure 32. GPIO Block Diagram(Except PC13, PC14, PC15) 102
Figure 33. PC13, PC14, PC15 Block Diagram 102
Figure 34. I/O Port Block Diagram (External Interrupt I/O Pins) 102
Figure 35. Port Diagram 116
Figure 36. Debounce Function Timing Diagram 117
Figure 37. GPIO Diagram 117
Figure 38. Flash Memory Map (128 KB Code Flash) 119
Figure 39. DMAC Block Diagram 129
Figure 40. DMAC Functional Block Diagram 135
Figure 41. DMAC Operation Sequence 136
Figure 42. Timing Diagram of DMAC Transfer from Peripheral to Memory 137
Figure 43. Timing Diagram of DMAC Transfer from Memory to Peripheral 137
Figure 44. Timing Diagram Example of N DMAC Transfer 137
Figure 45. Static Memory Controller Block Diagram 139
Figure 46. Bus Configuration with 8-bit Memory 142
Figure 47. Bus Configuration with 16-bit Memory 143

Figure 48. Word Read Operation with 8-bit Memory and 8-bit Bus 144
Figure 49. Half Word Read Operation with 8-bit Memory and 8-bit Bus 144
Figure 50. Word Read Operation with 8-bit Memory and 16-bit Bus 145
Figure 51. Half Word Read Operation with 8-bit Memory and 16-bit Bus 145
Figure 52. Word Read Operation with 16-bit Memory and 16-bit Bus 146
Figure 53. Half Word Read Operation with 16-bit Memory and 16-bit Bus 146
Figure 54. Word Write Operation with 8-bit Memory and 8-bit Bus 147
Figure 55. Half Word Write Operation with 8-bit Memory and 8-bit Bus 147
Figure 56. Word Write Operation with 8-bit Memory and 16-bit Bus 148
Figure 57. Half Word Write Operation with 8-bit Memory and 16-bit Bus 148
Figure 58. Word Write Operation with 16-bit Memory and 16-bit Bus 149
Figure 59. Half Word Write Operation with 16-bit Memory and 16-bit Bus 149
Figure 60. Read Timing Changing by 'IDLE = 1' 150
Figure 61. Write Timing Changing by 'PREIDL = 1' 150
Figure 62. Write Operation with External Wait (nWAIT), 'EXP = 0' 151
Figure 63. WDT Block Diagram 152
Figure 64. WDT Interrupt and WDT Reset Timing Diagram 158
Figure 65. Watch Timer Block Diagram 159
Figure 66. 16-bit Timer Block Diagram 164
Figure 67. TIMER 1n Block Diagram in Timer/Counter Mode (n = 0, 1, 2, and 3) 170
Figure 68. Timer/Counter Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3) 170
Figure 69. Timer/Counter Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3) 171
Figure 70. TIMER 1n Block Diagram in Capture Mode (n = 0, 1, 2, and 3) 172
Figure 71. Capture Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3) 173
Figure 72. Express Timer Overflow in Capture Mode of TIMER 1n (n = 0, 1, 2, and 3) 173
Figure 73. Capture Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3) 174
Figure 74. TIMER 1n Block Diagram in PPG Mode (n = 0, 1, 2, and 3) 175
Figure 75. PPG Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3) 176
Figure 76. PPG Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3) 177
Figure 77. 16-bit Timer Block Diagram 179
Figure 78. TIMER 2n Block Diagram in Timer/Counter Mode 186
Figure 79. Timer/Counter Mode Timing Example of TIMER 2n 186
Figure 80. Timer/Counter Mode Operation Sequence of TIMER 2n (n = 0 or 1) 187
Figure 81. TIMER 2n Block Diagram in Capture Mode 188
Figure 82. Capture Mode Timing Example of TIMER 2n (n = 0 and 1) 188
Figure 83. Express Timer Overflow in Capture Mode of TIMER 2n (T2nCNCLR = 1'b1) 189
Figure 84. Express Timer Overflow in Capture Mode of TIMER 2n (T2nCNCLR = 1'b0) 189
Figure 85. Capture Mode Operation Sequence of TIMER 2n (n = 0 or 1) 190
Figure 86. TIMER 2n Block Diagram in PPG Mode 191
Figure 87. PPG Mode Timing Example of TIMER 2n (n = 0 or 1) 192
Figure 88. PPG Mode Operation Sequence of TIMER 2n (n = 0 or 1) 193
Figure 89. Timer Counter 30 Block Diagram 195
Figure 90. 16-bit Capture Mode of Timer 30 209
Figure 91. Capture Mode Operation Sequence of TIMER 30 210
Figure 92. Example of PWM at 4MHZ 212
Figure 93. Example of Changing the Period in Absolute Duty Cycle at 4MHz 212
Figure 94. Interval Mode Timing Chart With "DLYPOS = 0" 214
Figure 95. Interval Mode Timing Chart With "DLYPOS = 1 215
Figure 96. Back-to-Back Mode Timing Chart 217
Figure 97. Example of PWM External Synchronization with BLNK Input (x: A, B and C) 218

Figure 98. Example of Force A-Channel Mode 219
Figure 99. Force A-Channel Mode Block Diagram 220
Figure 100. Example of 6-Channel Mode 221
Figure 101. 6-Channel Mode Block Diagram 222
Figure 102. Timer Counter 40 Block Diagram 224
Figure 103. TIMER 40 Block Diagram in 16-bit Timer/Counter Mode 231
Figure 104. 16-bit Timer/Counter Mode Timing Example of TIMER 40 231
Figure 105. 16-bit Timer/Counter Mode Operation Sequence of TIMER 40 232
Figure 106. 16-bit Capture Mode of Timer 40 233
Figure 107. Capture Mode Timing Example of TIMER 40 234
Figure 108. Express Timer Overflow in Capture Mode of TIMER 40 234
Figure 109. Capture Mode Operation Sequence of TIMER 40 235
Figure 110. TIMER 40 Block Diagram in PPG Mode 236
Figure 111. PPG Mode Timing Example of TIMER 40 237
Figure 112. PPG Mode Operation Sequence of TIMER 40 238
Figure 113. UART Block Diagram (n = 10, 11, 12, and 13) 240
Figure 114. SPIN Block Diagram (n = 10, 11, 12, and 13) 241
Figure 115. Clock Generation Block Diagram (USARTn, n = 10, 11, 12, and 13) 249
Figure 116. Synchronous Mode SCKn Timing (USARTn, n = 10, 11, 12, and 13) 250
Figure 117. Frame Format (USART) 251
Figure 118. Asynchronous Start Bit Sampling (n = 10, 11, 12 and 13) 256
Figure 119. Asynchronous Data and Parity Bit Sampling (n = 10, 11, 12 and 13) 256
Figure 120. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11, 12 and 13) 257
Figure 121. USART SPIn Clock Formats when CPHAn = 0 (n = 10, 11, 12 and 13) 258
Figure 122. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11, 12 and 13) 259
Figure 123. 0% Error Baud Rate Diagram 261
Figure 124. I2C Block Diagram 263
Figure 125. I2C Bus Bit Transfer (n = 0 and 1) 270
Figure 126. START and STOP Condition (n = 0 and 1) 270
Figure 127. I2C Bus Data Transfer (n = 0 and 1) 271
Figure 128. I2C Bus Acknowledge (n = 0 and 1) 272
Figure 129. Clock Synchronization during the Arbitration Procedure (n = 0 and 1) 272
Figure 130. Arbitration Procedure between Two Masters (n = 0 and 1) 273
Figure 131. SPI Block Diagram 281
Figure 132. SPI wave form (STL, BTL and SPL) 287
Figure 133. SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0) 288
Figure 134. SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1) 288
Figure 135. SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0) 289
Figure 136. SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1) 289
Figure 137. DMA Handshake Flow Chart 290
Figure 138. 12-bit ADC Block Diagram 292
Figure 139. ADC Single Mode Timing (When ADC.MR.AMOD = '0') 304
Figure 140. ADC Burst Mode Timing (When ADC.MR.AMOD = '1') 305
Figure 141. ADC Trigger Timing in Burst Mode (SEQCNT = 3'b111, 8 Sequential Conversion) 305
Figure 142. ADC Sequential Mode Timing (When MR.AMOD = '0' and MR.SEQCNT ≠ '0') 306
Figure 143. ADC Trigger Timing in Sequential Mode (SEQCNT = 3'b111, 8 Sequential Conversion) 306
Figure 144. ADC Multiple Mode Timing (When MR.AMOD = '2' and MR.SEQCNT ≠ '0') 307
Figure 145. ADC Trigger Timing in Multiple Mode (SEQCNT = 3'b111, 8 Multiple Conversion) 307
Figure 146. 10-bit DAC Block Diagram 308
Figure 147. Analog Power (AVREF) Pin with a Capacitor 309

Figure 148. DACBR[15:0] Value by a Selected Gain 315
Figure 149. Comparator Block Diagram 316
Figure 150. CRC and Checksum Block Diagram 323
Figure 151. CRC Polynomial Structure 327
Figure 152. USB Interface Block Diagram 328
Figure 153. USB Packet Field 404
Figure 154. USB Descriptor Hierarchy 405
Figure 155. USB Communication using Software 409
Figure 156 Application Example: Open Windows OS 410
Figure 157. Application Example: Check COM Port 410
Figure 158. Application Example: Printing Characters 410
Figure 159. HID Block Diagram 411
Figure 160. USB HID PIPE Example GUI 411
Figure 161. Crystal Design Guide 412
Figure 162. RTC Block Diagram 414
Figure 163. Start Operation Sequence of RTC 428
Figure 164. Read Operation Sequence of RTC 429
Figure 165. Sub-second Counter Match in RW Mode 430
Figure 166. Sub-second Counter Match in RW Mode when Writing to SEC 430
Figure 167. Alarm Operation Sequence of RTC 431
Figure 168. Configuring Sequence of BKUPDR Register 435
Figure 169. PC13 wakeup in Standby Mode 436
Figure 170. Temp Sensor Block Diagram 437
Figure 171. AC Timing 448
Figure 172. SPI Timing 450
Figure 173. I2C Timing 451
Figure 174. Waveform of UART Timing Characteristics 452
Figure 175. UART Module Timing 452
Figure 176. Crystal/Ceramic Oscillator 455
Figure 177. External Clock 455
Figure 178. Clock Timing Measurement at XIN 455
Figure 179. Crystal Oscillator 456
Figure 180. Clock Timing Measurement at SXIN 457
Figure 181. Operating Voltage Range 457
Figure 182. A-Link and Pin Descriptions 461
Figure 183. E-PGM+ (Single Writer) and Pin Descriptions 462
Figure 184. E-Gang4 and E-Gang6 (for Mass Production) 462
Figure 185. PCB Design Guide for On-Board Programming 463
Figure 186. 64 LQFP Package Outline 464
Figure 187. 48 LQFP Package Outline 466
Figure 187. 48 QFN Package Outline 468
Figure 188. A31G32x Device Numbering Nomenclature 470

List of tables

Table 1. A31G324 and A31G323 Device Features and Peripheral Counts 19
Table 2. Pin Description 26
Table 3. Interrupt Vector Map 35
Table 4. A31G32x Memory Boundary Addresses 38
Table 5. Boot Mode Pin List 40
Table 6. SCU Pins 43
Table 7. Clock Sources 45
Table 8. Flash Wait Control Recommendation 47
Table 9. Peripheral Clock Select 48
Table 10. Reset Sources of Cold Reset and Warm Reset 49
Table 11. Operation Mode 54
Table 12. Oscillator Control in Power down Mode 56
Table 13. Peripheral Operation in Power down Mode 56
Table 14. STOP Mode Configuration 57
Table 15. STANDBY Mode Configuration 59
Table 16. BACKUP POWER Mode Configuration 61
Table 17. Base Address of SCU (Chip Configuration) 62
Table 18. SCU Register Map (Chip Configuration) 62
Table 19. Base Address of SCU 63
Table 20. SCU Register Map 63
Table 21. Base Address of LVI/LVR 64
Table 22. LVI/LVR Register Map 64
Table 23. VDC33 Mode Description 88
Table 24. PCU and GPIO pins 101
Table 25. GPIO Alternative Function 103
Table 26. Base Address of PCU 105
Table 27. PCU and GPIO Register Map 105
Table 28. Flash Memory Controller Features 118
Table 29. Base Address of Flash Memory Controller 120
Table 30. FMC Register Map 120
Table 31. Base Address of DMAC 130
Table 32. DMAC Register Map 130
Table 33. DMAC PERISEL Selection 131
Table 34. Pin Assignment of Static Memory Controller 138
Table 35. Base Address of Static Memory Controller 139
Table 36. SMIF Register Map 139
Table 37. Memory Region Selection and nCS Output by Address 141
Table 38. Data Sort for 8-bit Memory Bus 141
Table 39. Data Sort for 16-bit Memory Bus 142
Table 40. Config Signal by A[0] 142
Table 41. Base Address of WDT 153
Table 42. WDT Register Map 153
Table 43. Pre-scaled WDT Counter Clock Frequency 158
Table 44. Base Address of WT 160
Table 45. WT Register Map 160
Table 46. Pin Assignment of 16-bit Timer: External Pins 163
Table 47. Base Address of 16-bit Timer 164
Table 48. TIMER 1n Register Map 165

Table 49. TIMER 1n Operating Modes 169
Table 50. Pin Assignment of 32-bit Timer: External Pins 178
Table 51. Base Address of 32-bit Timer 180
Table 52. TIMER 2n Register Map 180
Table 53. TIMER 2n Operating Modes 185
Table 54. Pin Assignment of Timer Counter 30: External Pins 194
Table 55. Base Address of Timer Counter 30 196
Table 56. Timer Counter 30 Register Map 196
Table 57. Timer 30 Operating Modes 208
Table 58. PWM Channel Polarity 211
Table 59. Pin Assignment of Timer Counter 40: External Pins 223
Table 60. Base Address of Timer Counter 40 225
Table 61. Timer Counter 30 Register Map 225
Table 62. Timer 40 Operating Modes 230
Table 63. Pin Assignment of USART: External Pins 239
Table 64. Base Address of USART 241
Table 65. USART Register Map 242
Table 66. Equations for Calculating USART Baud Rate Register Settings 249
Table 67. CPOL Functionality 258
Table 68. Baud Rate Settings Example 260
Table 69. Pin Assignment of I2C: External Pins 262
Table 70. Base Address of I2C Interface 263
Table 71. I2C Register Map 264
Table 72. Pin Assignment of SPI: External Pins 280
Table 73. Base Address of SPI 281
Table 74. SPI Register Map 281
Table 75. Pin Assignment of ADC: External Signal 291
Table 76. Base Address of 12-bit ADC 292
Table 77. 12-bit ADC Register Map 292
Table 78. Trigger Source of ADC 298
Table 79. Base Address of DAC 309
Table 80. 10-bit DAC Register Map 309
Table 81. Base Address of Comparator 317
Table 82. Comparator Register Map 317
Table 83. Base Address of CRC 324
Table 84. CRC Register Map 324
Table 85. Base Address of USB Interface 329
Table 86. USB Interface Register Map 329
Table 87. USB Packet PID 404
Table 88. Base Address of RTC Interface 415
Table 89. RTC Register Map 415
Table 90. Specified time system by AMPM 420
Table 91. SUBDR and RTC Clock Combination for 1Hz 427
Table 92. WEEK and ALARMWW Registers 431
Table 93. Example of Setting the Alarm 432
Table 94. Correction Resolution by Frequency 433
Table 95. PC13, PC14, PC15 Function in RTCPFCR register 436
Table 96. Base Address of TSENSE Interface 438
Table 97. TSENSE Register Map 438
Table 98. Specific temperature value in the region of OTP 440

Table 99. Absolute maximum rating 441
Table 100. Recommended Operating Condition 442
Table 101. ADC Electrical Characteristics 442
Table 102. DAC Electrical Characteristics 443
Table 103. POR Electrical Characteristics 443
Table 104. Low Voltage Reset/Indicator Characteristics 444
Table 105. High Frequency Internal RC Oscillator Characteristics 445
Table 106. Low Frequency (750KHz) Internal RC Oscillator Characteristics 445
Table 107. Low Frequency (40KHz) Internal RC Oscillator Characteristics 445
Table 108. DC Electrical Characteristics 446
Table 109. Supply Current Characteristics 447
Table 110. AC Characteristics 448
Table 111. SPI Characteristics with High Voltage 449
Table 112. SPI Characteristics with Low Voltage 449
Table 113. I2C Characteristics 451
Table 114. UART Timing Characteristics 452
Table 115. Data Retention Voltage in STOP mode 453
Table 116. Internal Flash ROM Characteristics 453
Table 117. Main Oscillator Characteristics 454
Table 118. Sub Oscillator Characteristics 456
Table 119. PLL Electrical Characteristics 457
Table 120. Comparator Characteristics 458
Table 121. Interanal Voltage Reference Characteristics(For comparator) 459
Table 122. USB Characteristics 460
Table 123. Temperature Sensor Characteristics 460
Table 124. 64 LQFP Package Mechanical Data 465
Table 125. 48 LQFP Package Mechanical Data 467
Table 125. 48 QFN Package Mechanical Data 469
Table 126. A31G32x Device Ordering Information 470

1 Description

A31G32x lines are 32-bit general purpose microcontrollers with up to 128 Kbytes of flash memory. They are powerful microcontrollers which provide effective solutions to various electrical appliances which require both low power consumption and high performance.

1.1 Device overview

In this section, features of A31G32x and peripheral counts are introduced.

Table 1. A31G324 and A31G323 Device Features and Peripheral Counts

Peripherals		Description
Core	CPU	High Performance Low-Power Cortex-M0+ Core 32-bit ARM Cortex-M0+ CPU
	Interrupt	NVIC (Nested-Vectored Interrupt Controller) Up to 32 peripheral interrupts supported.
Memory	Code flash	A31G324: 128Kbytes code flash memory A31G323: 64Kbytes code flash memory Flash access wait 0 clock wait: up to 20MHz 1 clock wait: up to 40MHz 2 clock wait: up to 48MHz
	SRAM	16 Kbytes SRAM (0 wait)
	Endurance	10,000 times at room temperature Retention for 10 years
System Control Unit (SCU)	Operating frequency	Up to 48 MHz
	Clock	High speed internal oscillator (HSI) 48MHz (±3% @-40℃ to +85℃) Low speed internal oscillator (LSI) 750KHz (±20% @-40℃ to +85℃) 40KHz (±50% @-40℃ to +85℃) External main oscillator (HSE): 2MHz to 16MHz External sub-oscillator (LSE): 32.768KHz Phase-locked loop (PLL) frequency generator generates a high-speed clock (up to 144MHz)
	Clock monitoring	System Fail-Safe function by Clock Monitoring External main oscillator(HSE) External sub oscillator(LSE) Main system clock (MCLK)

Table 1. A31G324 and A31G323 Device Features and Peripheral Counts (continued)

Peripherals		Description
	Operating mode	RUN mode SLEEP mode STOP mode STANDBY mode Backup Power mode
	Reset	nRESET pin reset Core reset Software reset POR (Power On Reset) LVR (Low Voltage Reset) WDTR (Watch Dog Timer Reset) Reset due to clock oscillating error
General Purpose I/O (GPIO)		51 Ports (PA[15:0], PB[15:0], PC[15:0], PF[2:0]): 64-Pin 37 Ports (PA[15:0], PB[15:0], PC[15:13], PF[1:0]): 48-Pin
Direct Memory Access Controller (DMA)		4-ch direct memory access (DMA) support peripherals SPI20, SPI21, USART10, USART11, USART12, USART13, CRC, ADC
Watch Timer (WT)		14-bit divider with extended 12-bit counter 1-ch
Watchdog Timer (WDT)		24-bit down counter timer: 1-ch Reset and periodic interrupts are supported
TIMER	Timer1x	16bit: 4-ch Periodic timer mode, One-shot timer mode, PWM pulse mode, Capture mode
	Timer2x	32bit : 2-ch Periodic timer mode, One-shot timer mode, PWM pulse mode, Capture mode
	Timer40	16bit : 1-ch Periodic timer mode, One-shot timer mode, PWM pulse mode, 3-Phase Capture mode
PWM	Timer30	16bit : 6-ch Periodic timer mode, Back-to-Back mode, Capture mode

Table 1. A31G324 and A31G323 Device Features and Peripheral Counts (continued)

Peripherals		Description
Communication function	USART	4-ch
	SPI	2-ch
	I2C	2-ch
	USB	Full speed 2.0 device
ADC		12-bit ADC : 1Msps 64-pin : 16-ch 48-pin : 10-ch
DAC		10-bit DAC 1-ch PA4, PA5: output pin
External Bus Interface (EBI)		20-bit address size, 16-bit multiplexed address/data bus
CRC calculator (CRC)		CRC-CCITT, CRC-16
Comparator		8-ch reference input 2-ch output comparator Window mode
Operating Voltage		1.8V to 5.5V
Operating temperature		Commercial grade (-40℃ to +85℃)
Package		Three types of package options 48-pin QFN 48-pin LQFP 64-pin LQFP

1.2 Block diagram

In this section, A31G32x devices with peripherals are described in block diagram.

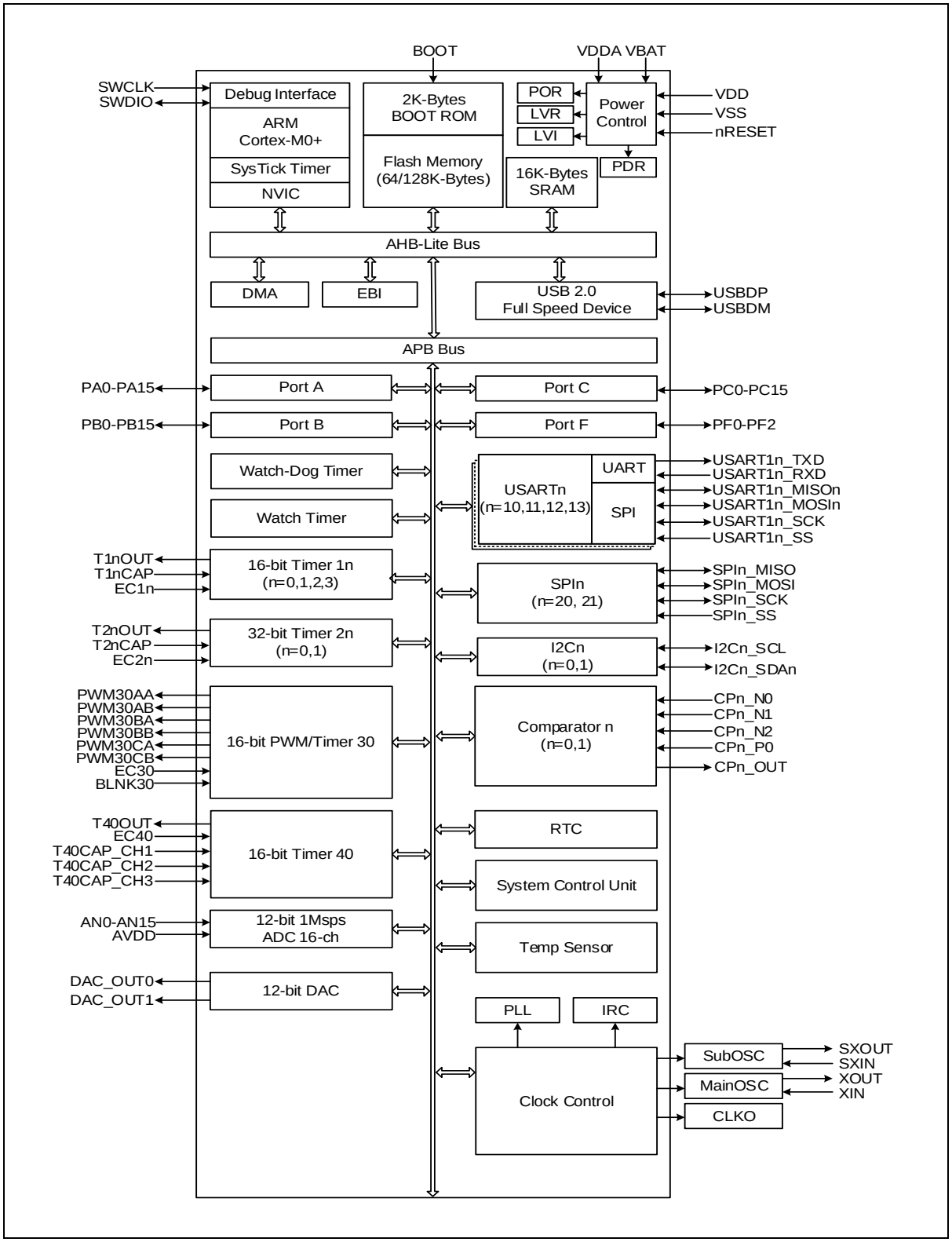


Figure 1. A31G32x Block diagram

2. Pinouts and pin descriptions

In this chapter, A31G32x devices' pinouts and pin descriptions are introduced.

2.1 Pinouts

2.1.1 A31G323RLN, A31G324RLN (64 LQFP)

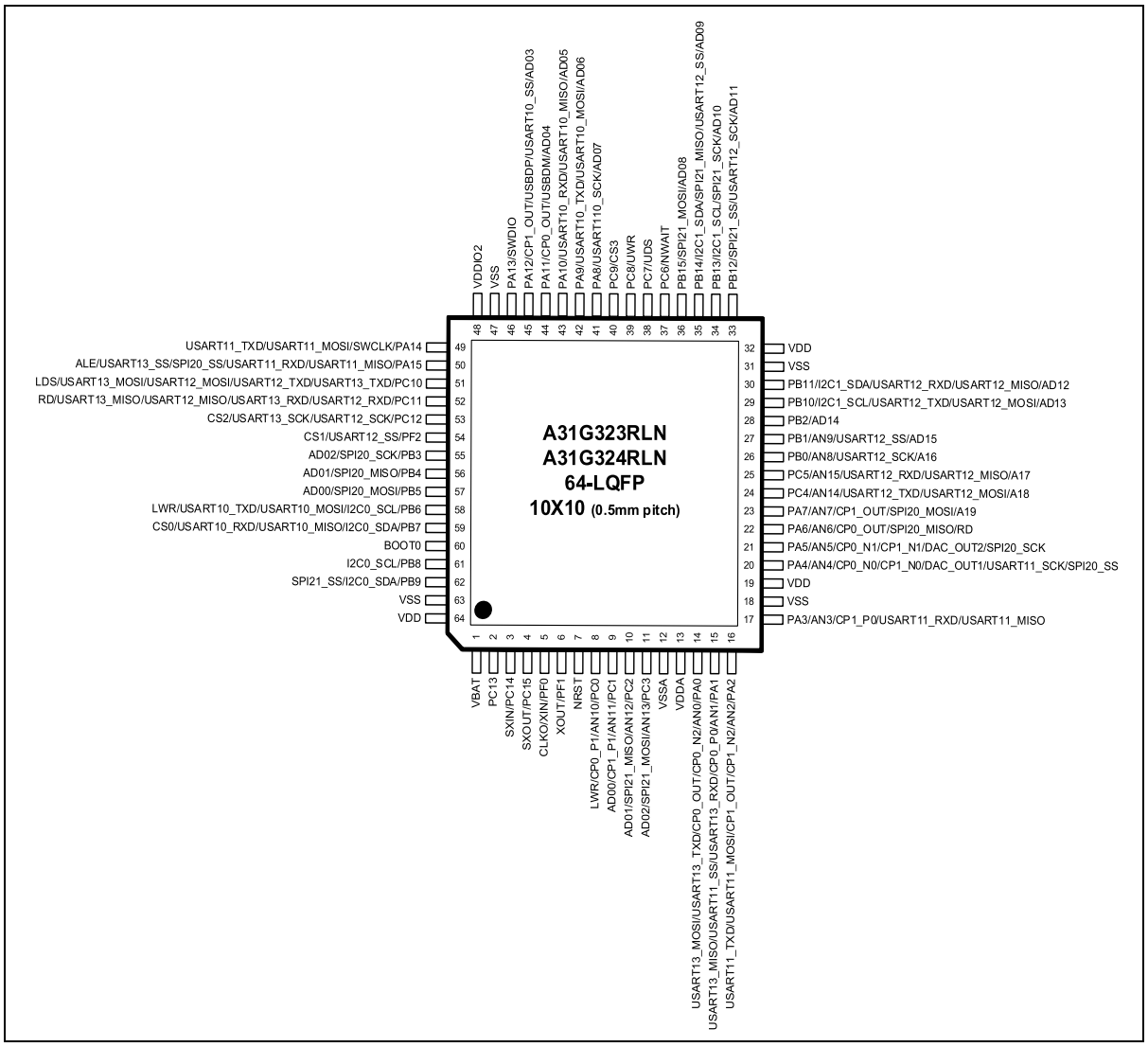


Figure 2. LQFP 64 Pinouts

2.1.2 A31G323CLN, A31G324CLN (48 LQFP)

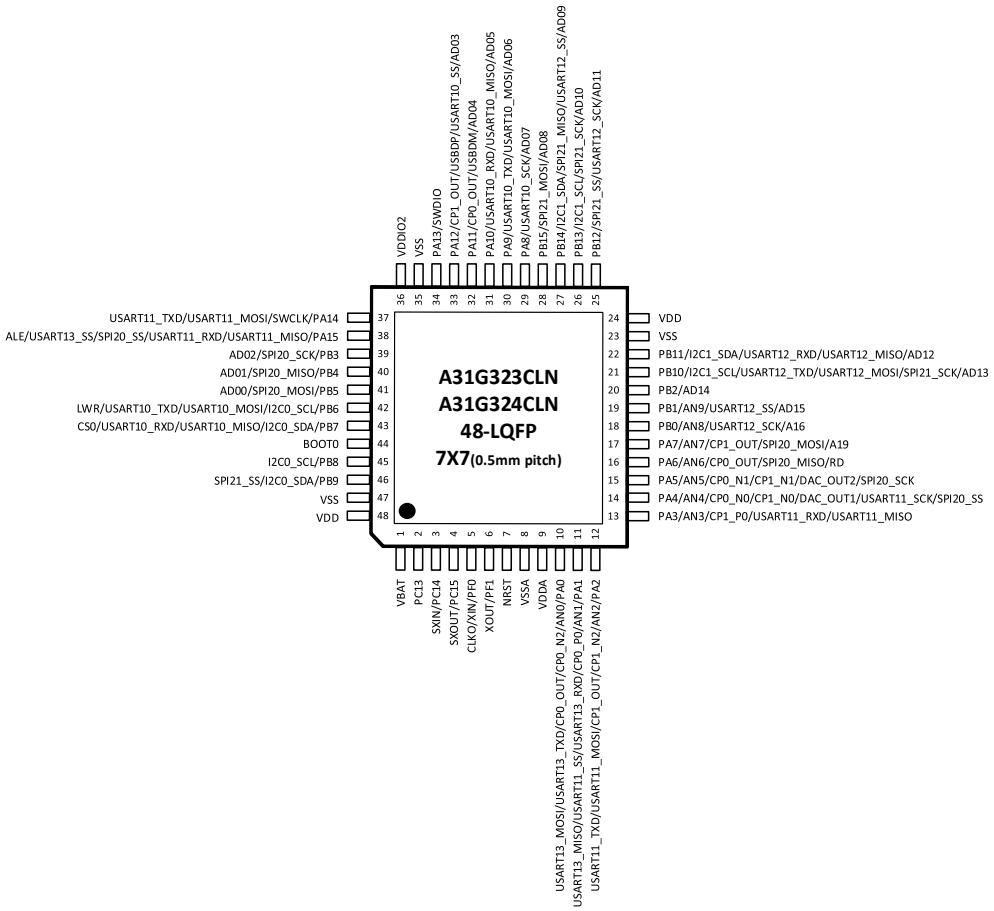
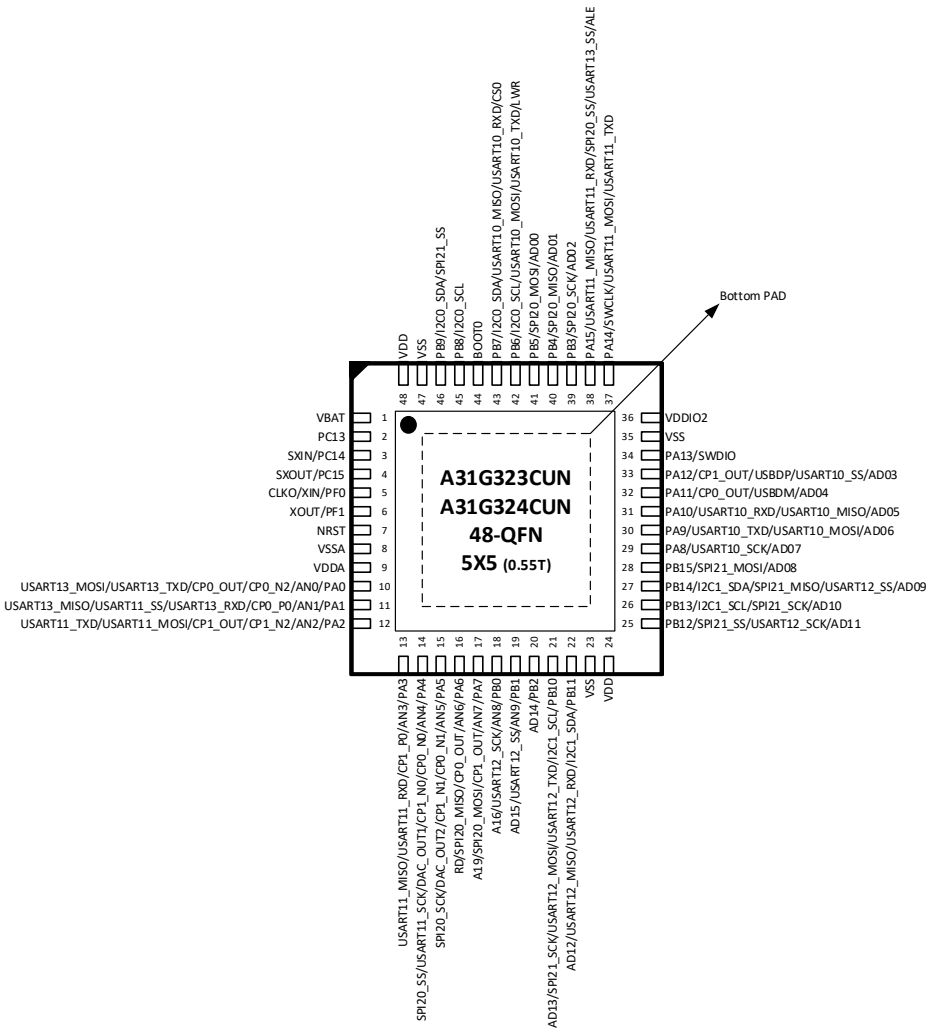


Figure 3. LQFP 48 Pinouts

2.1.3 A31G323CUN, A31G324CUN (48 QFN)



NOTE

1. Bottom PAD is Floating. (Connectable to VSS)

Figure 4. QFN 48 Pinouts

2.2 Pin description

Pin configuration information in Table 2 contains two pairs of power/ground and other dedicated pins. These multi-function pins have up to ten selections of functions including GPIO. Configuration including pin ordering can be changed without notice.

Table 2. Pin Description

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
1	1	VBAT	P	Backup Power Supply	
2	2	PC13*	IOUDS	PORT C Bit 13 Input/Output	
		RTC_TAMP1	I	RTC TimeStamp Input	
		RTC_OUT	O	RTC Output	
3	3	PC14*	IOUDS	PORT C Bit 14 Input/Output	
		SXIN	I	Sub Oscillator Input	
4	4	PC15*	IOUDS	PORT C Bit 15 Input/Output	
		SXOUT	O	Sub Oscillator Output	
5	5	PF0*	IOUDS	PORT F Bit 0 Input/Output	
		XIN	I	Main Oscillator Input	
		CLKO	O	Clock Output	
6	6	PF1*	IOUDS	PORT F Bit 1 Input/Output	
		XOUT	O	Main Oscillator Output	
7	7	nReset	IU	Reset Pin	Pull-up
8	__	PC0*	IOUDS	PORT C Bit 0 Input/Output	
		AN10	IA	ADC Input 10	
		LWR	O	EBI Lower byte write signal	
		CP0_P1	IA	Comparator 0 Positive 1 Input	
9	__	PC1*	IOUDS	PORT C Bit 1 Input/Output	
		AN11	IA	ADC Input 11	
		AD00	I/O	EBI Address/Data bus bit 0	
		CP1_P1	IA	Comparator 1 Positive 1 Input	
10	__	PC2*	IOUDS	PORT C Bit 2 Input/Output	
		SPI21_MISO	I/O	SPI Channel 21 Master Input/Slave Output	
		AN12	IA	ADC Input 12	
		AD01	I/O	EBI Address/Data bus bit 1	
11	__	PC3*	IOUDS	PORT C Bit 3 Input/Output	
		SPI21_MOSI	I/O	SPI Channel 21 Master Out / Slave In	
		AN13	IA	ADC Input 13	
		AD02	I/O	EBI Address/Data bus bit 2	
		EC13	I	Timer 13 Event Count Input	
		T13CAP	I	Timer 13 Capture Input	
		T13OUT	O	Timer 13 Output	
12	8	AVSS	P	Power supply for ADC, DAC, Comparator	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
13	9	AVDD	P	Power supply for ADC, DAC, Comparator	
14	10	PA0*	IOUDS	PORT A Bit 0 Input/Output	
		CP0_OUT	O	Comparator 0 Output	
		USART13_TXD	O	USART Channel 13 TXD Output	
		USART13_MOSI	I/O	USART Channel 13 Master Out / Slave In	
		AN0	IA	ADC Input 0	
		CP0_N2	IA	Comparator 0 Negative 2 Input	
		RTC_TAMP2	I	RTC TimeStamp Input	
		EC20	I	Timer 20 External Clock Input	
15	11	PA1*	IOUDS	PORT A Bit 1 Input/Output	
		USART11_SS	I/O	USART Channel 11 Slave Select Signal	
		USART13_RXD	I	USART Channel 13 RXD Input	
		USART13_MISO	I/O	USART Channel 13 Master Input/Slave Output	
		AN1	IA	ADC Input 1	
		CP0_P0	IA	Comparator 0 Positive 0 Input	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		T13OUT	O	Timer 13 Output	
		T13CAP	I	Timer 13 Capture Input	
		EC13	I	Timer 13 External Clock Input	
16	12	PA2*	IOUDS	PORT A Bit 2 Input/Output	
		USART11_TXD	O	USART Channel 11 TXD Output	
		USART11_MOSI	I/O	USART Channel 11 Master Output/Slave Input	
		CP1_OUT	O	Comparator 1 Output	
		AN2	IA	ADC Input 2	
		CP1_N2	IA	Comparator 1 Negative 2 Input	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		T10OUT	O	Timer 10 Output	
		T10CAP	I	Timer 10 Capture Input	
		EC10	I	Timer 10 External Clock Input	
17	13	PA3*	IOUDS	PORT A Bit 3 Input/Output	
		USART11_RXD	I	USART Channel 11 RXD Input	
		USART11_MISO	I/O	USART Channel 11 Master Input/Slave Output	
		AN3	IA	ADC Input 3	
		CP1_P0	IA	Comparator 1 Positive 0 Input	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		T10OUT	O	Timer 10 Output	
		T10CAP	I	Timer 10 Capture Input	
		EC10	I	Timer 10 External Clock Input	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
18	__	VSS	P	VSS	
19	__	VDD	P	VDD	
20	14	PA4*	IOUDS	PORT A Bit 4 Input/Output	
		SPI20_SS	I/O	SPI Channel 20 Slave Select Signal	
		CP0_N0	IA	Comparator 0 Negative 0 Input	
		CP1_N0	IA	Comparator 1 Negative 0 Input	
		AN4	IA	ADC Input 4	
		USART11_SCK	I/O	USART Channel 11 Clock Input/Output	
		DAC_OUT1	O	DAC Output	
21	15	PA5*	IOUDS	PORT A Bit 5 Input/Output	
		SPI20_SCK	I/O	SPI Channel 20 Clock Input/Output	
		CP0_N1	IA	Comparator 0 Negative 1 Input	
		CP1_N1	IA	Comparator 1 Negative 1 Input	
		AN5	IA	ADC Input 5	
		EC20	I	Timer 20 External Clock Input	
		DAC_OUT2	O	DAC Output	
22	16	PA6*	IOUDS	PORT A Bit 6 Input/Output	
		SPI20_MISO	I/O	SPI Channel 20 Master-Input/Slave-Output	
		CP0_OUT	O	Comparator 0 Output	
		AN6	IA	ADC Input 6	
		BLNK30	I	External Sync Signal Input for T30 PWM	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH1	I	Timer 40 CH1 Capture Input	
		T11OUT	O	Timer 11 Output	
		T11CAP	I	Timer 11 Capture Input	
		EC11	I	Timer 11 External Clock Input	
		RD	O	EBI Read signal	
23	17	PA7*	IOUDS	PORT A Bit 7 Input/Output	
		SPI20_MOSI	I/O	SPI Channel 20 Master Out / Slave In	
		CP1_OUT	O	Comparator 1 Output	
		AN7	IA	ADC Input 7	
		PWM30AB	O	Timer 30 PWM Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH2	I	Timer 40 CH2 Capture Input	
		T12OUT	O	Timer 12 Output	
		T12CAP	I	Timer 12 Capture Input	
		EC12	I	Timer 12 External Clock Input	
		A19	O	EBI Address 19 bus signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
24	__	PC4*	IOUDS	PORT C Bit 4 Input/Output	
		USART12_TXD	O	USART Channel 12 TXD Input	
		USART12_MOSI	I/O	USART Channel 12 Master Output/Slave Input	
		AN14	IA	ADC Input 14	
		EC40	I	Timer 40 External Clock Input	
		A18	O	EBI Address 18 bus signal	
25	__	PC5*	IOUDS	PORT C Bit 5 Input/Output	
		USART12_RXD	I	USART Channel 12 RXD Input	
		USART12_MISO	I/O	USART Channel 12 Master Input/Slave Output	
		AN15	IA	ADC Input 15	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		A17	O	EBI Address 17 bus signal	
26	18	PB0*	IOUDS	PORT B Bit 0 Input/Output	
		AN8	IA	ADC Input 8	
		PWM30BB	O	Timer 30 PWM Output	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH3	I	Timer 40 CH3 Capture Input	
		USART12_SCK	I/O	USART Channel 12 Clock Input/Output	
		A16	O	EBI Address 16 bus signal	
27	19	PB1*	IOUDS	PORT B Bit 1 Input/Output	
		AN9	IA	ADC Input 9	
		PWM30CB	O	Timer 30 PWM Output	
		USART12_SS	I/O	USART Channel 12 Slave Select Signal	
		T13OUT	O	Timer 13 Output	
		T13CAP	I	Timer 13 Capture Input	
		EC13	I	Timer 13 External Clock Input	
		AD15	I/O	EBI Address or Data 15 bus signal	
28	20	PB2*	IOUDS	PORT B Bit 2 Input/Output	
		T13OUT	O	Timer 13 Output	
		T13CAP	I	Timer 13 Capture Input	
		EC13	I	Timer 13 External Clock Input	
		AD14	I/O	EBI Address or Data 14 bus signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
29	21	PB10*	IOUDS	PORT B Bit 10 Input/Output	
		SPI21_SCK	I/O	SPI Channel 21 Clock Input/Output	
		I2C1_SCL	I/O	I2C Channel 1 SCL In/Output	
		USART12_TXD	O	USART Channel 12 TXD Input	
		USART12_MOSI	I/O	USART Channel 12 Master Output/Slave Input	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		AD13	I/O	EBI Address or Data 13 bus signal	
30	22	PB11*	IOUDS	PORT B Bit 11 Input/Output	
		USART12_RXD	I	USART Channel 12 RXD Input	
		USART12_MISO	I/O	USART Channel 12 Master Input/Slave Output	
		I2C1_SDA	I/O	I2C Channel 1 SDA In/Output	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		AD12	I/O	EBI Address or Data 12 bus signal	
31	23	VSS	P	VSS	
32	24	VDD	P	VDD	
33	25	PB12*	IOUDS	PORT B Bit 12 Input/Output	
		SPI21_SS	I/O	SPI Channel 21 Slave Select Signal	
		BLNK30	I	External Sync Signal Input for T30 PWM	
		USART12_SCK	I/O	USART Channel 12 Clock Input/Output	
		EC13	I	Timer 13 External Clock Input	
		T13CAP	I	Timer 13 Capture Input	
		T13OUT	O	Timer 13 Output	
		AD11	I/O	EBI Address or Data 11 bus signal	
34	26	PB13*	IOUDS	PORT B Bit 13 Input/Output	
		SPI21_SCK	I/O	SPI Channel 21 Clock Input/Output	
		I2C1_SCL	I/O	I2C Channel 1 SCL In/Output	
		PWM30AB	O	Timer 30 PWM Output	
		AD10	I/O	EBI Address or Data 10 bus signal	
35	27	PB14*	IOUDS	PORT B Bit 14 Input/Output	
		SPI21_MISO	I/O	SPI Channel 21 Master-Input/Slave-Output	
		I2C1_SDA	I/O	I2C Channel 1 SDA In/Output	
		PWM30BB	O	Timer 30 PWM Output	
		T10OUT	O	Timer 10 Output	
		T10CAP	I	Timer 10 Capture Input	
		EC10	I	Timer 10 External Clock Input	
		USART12_SS	I/O	USART Channel 12 Slave Select Signal	
		AD09	I/O	EBI Address or Data 9 bus signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
36	28	PB15*	IOUDS	PORT B Bit 15 Input/Output	
		SPI21_MOSI	I/O	SPI Channel 21 Master Out / Slave In	
		PWM30CB	O	Timer 30 PWM Output	
		T10OUT	O	Timer 10 Output	
		T10CAP	I	Timer 10 Capture Input	
		EC10	I	Timer 10 External Clock Input	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		AD08	I/O	EBI Address or Data 8 bus signal	
37	—	PC6*	IOUDS	PORT C Bit 6 Input/Output	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH1	I	Timer 40 CH1 Capture Input	
		nWAIT	O	EBI External Wait signal	
38	—	PC7*	IOUDS	PORT C Bit 7 Input/Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH2	I	Timer 40 CH2 Capture Input	
		UDS	O	EBI Upper Data select signal	
39	—	PC8*	IOUDS	PORT C Bit 8 Input/Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH3	I	Timer 40 CH3 Capture Input	
		UWR	O	EBI Upper Byte write signal	
40	—	PC9*	IOUDS	PORT C Bit 9 Input/Output	
		EC40	I	Timer 40 External Clock Input	
		CS3	O	EBI memory selection signal 3	
41	29	PA8*	IOUDS	PORT A Bit 8 Input/Output	
		USART10_SCK	I/O	USART Channel 10 Clock Input/Output	
		PWM30AA	O	Timer 30 PWM Output	
		AD07	I/O	EBI Address or Data 7 bus signal	
42	30	PA9*	IOUDS	PORT A Bit 9 Input/Output	
		USART10_TXD	O	USART Channel 10 TXD Input	
		USART10_MOSI	I/O	USART Channel 10 Master Output/Slave Input	
		PWM30BA	O	Timer 30 PWM Output	
		AD06	I/O	EBI Address or Data 6 bus signal	
43	31	PA10*	IOUDS	PORT A Bit 10 Input/Output	
		USART10_RXD	I	USART Channel 10 RXD Input	
		USART10_MISO	I/O	USART Channel 10 Master Input/Slave Output	
		PWM30CA	O	Timer 30 PWM Output	
		EC21	I	Timer 21 External Clock Input	
		AD05	I/O	EBI Address or Data 5 bus signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
44	32	PA11*	IOUDS	PORT A Bit 11 Input/Output	
		CP0_OUT	O	Comparator 0 Output	
		USBDM	I/O	USB Data Line Minus	
		AD04	I/O	EBI Address or Data 4 bus signal	
45	33	PA12*	IOUDS	PORT A Bit 12 Input/Output	
		CP1_OUT	O	Comparator 2 Output	
		USBDP	I/O	USB Data Line Plus	
		EC30	I	Timer 30 External Clock Input	
		USART10_SS	I/O	USART Channel 10 Slave Select Signal	
		T30CAP	I	Timer 30 Capture Input	
		AD03	I/O	EBI Address or Data 3 bus signal	
46	34	PA13	IOUDS	PORT A Bit 13 Input/Output	
		SWDIO*	I/OU	SWD Data Input/Output	Pull-up
47	35	VSS	P	VSS	
48	36	VDDIO2	P	VDD	
49	37	PA14	IOUDS	PORT A Bit 14 Input/Output	
		USART11_TXD	O	USART Channel 11 TXD Output	
		USART11_MOSI	I/O	USART Channel 11 Master Output/Slave Input	
		SWCLK*	IU	SWD Clock Input	Pull-up
50	38	PA15*	IOUDS	PORT A Bit 15 Input/Output	
		USART13_SS	I/O	USART Channel 13 Slave Select Signal	
		USART11_RXD	I	USART Channel 11 RXD Input	
		USART11_MISO	I/O	USART Channel 11 Master Input/Slave Output	
		SPI20_SS	I/O	SPI Channel 20 Slave Select Signal	
		EC21	I	Timer 21 External Clock Input	
		ALE	O	EBI Address Latch signal	
51	—	PC10*	IOUDS	PORT C Bit 10 Input/Output	
		USART12_TXD	O	USART Channel 12 TXD Output	
		USART12_MOSI	I/O	USART Channel 12 Master Output/Slave Input	
		USART13_TXD	O	USART Channel 13 TXD Output	
		USART13_MOSI	I/O	USART Channel 13 Master Output/Slave Input	
		LDS	O	EBI Lower Data select signal	
52	—	PC11*	IOUDS	PORT C Bit 11 Input/Output	
		USART12_RXD	I	USART Channel 12 RXD Input	
		USART12_MISO	I/O	USART Channel 12 Master Input/Slave Output	
		USART13_RXD	I	USART Channel 13 RXD Input	
		USART13_MISO	I/O	USART Channel 13 Master Input/Slave Output	
		EC21	I	Timer 21 External Clock Input	
		RD	O	EBI Read signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
53	—	PC12*	IOUDS	PORT C Bit 12 Input/Output	
		USART12_SCK	I/O	USART Channel 12 Clock Input/Output	
		USART13_SCK	I/O	USART Channel 13 Clock Input/Output	
		CS2	O	EBI memory selection signal 2	
54	—	PF2*	IOUDS	PORT F Bit 2 Input/Output	
		USART12_SS	I/O	USART Channel 12 Slave Select Signal	
		EC40	I	Timer 40 External Clock Input	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		CS1	O	EBI memory selection signal 1	
55	39	PB3*	IOUDS	PORT B Bit 3 Input/Output	
		SPI20_SCK	I/O	SPI Channel 20 Clock Input/Output	
		T20OUT	O	Timer 20 Output	
		T20CAP	I	Timer 20 Capture Input	
		AD02	I/O	EBI Address or Data 2 bus signal	
56	40	PB4*	IOUDS	PORT B Bit 4 Input/Output	
		SPI20_MISO	I/O	SPI Channel 20 Master-Input/Slave-Output	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH1	I	Timer 40 CH1 Capture Input	
		AD01	I/O	EBI Address or Data 1 bus signal	
57	41	PB5*	IOUDS	PORT B Bit 5 Input/Output	
		SPI20_MOSI	I/O	SPI Channel 20 Master Out / Slave In	
		T40OUT	O	Timer 40 Output	
		T40CAP_CH2	I	Timer 40 CH2 Capture Input	
		T21CAP	I	Timer 21 Capture Input	
		T21OUT	O	Timer 21 Output	
		AD00	I/O	EBI Address or Data 0 bus signal	
58	42	PB6*	IOUDS	PORT B Bit 6 Input/Output	
		I2C0_SCL	O	I2C Channel 0 SCL In/Output	
		USART10_TXD	O	USART Channel 10 TXD Output	
		USART10_MOSI	I/O	USART Channel 10 Master Output/Slave Input	
		T11OUT	O	Timer 11 Output	
		T11CAP	I	Timer 11 Capture Input	
		EC11	I	Timer 11 External Clock Input	
		T40CAP_CH3	I	Timer 40 CH3 Capture Input	
		T40OUT	O	Timer 40 Output	
		LWR	O	EBI Lower Byte Write signal	

Table 2. Pin Description (continued)

Pin no.		Pin name	Type	Description	Remark
64-pin	48-pin				
59	43	PB7*	IOUDS	PORT B Bit 7 Input/Output	
		I2C0_SDA	I/O	I2C Channel 0 SDA In/Output	
		USART10_RXD	I	USART Channel 10 RXD Input	
		USART10_MISO	I/O	USART Channel 10 Master Input/Slave Output	
		T12OUT	O	Timer 12 Output	
		T12CAP	I	Timer 12 Capture Input	
		EC12	I	Timer 12 External Clock Input	
		CS0	O	EBI memory selection signal 0	
60	44	BOOT0	IU	Boot Mode Selection Input	Pull-up
61	45	PB8*	IOUDS	PORT B Bit 8 Input/Output	Pull-up
		I2C0_SCL	I/O	I2C Channel 0 SCL In/Output	
		T11OUT	O	Timer 11 Output	
		T11CAP	I	Timer 11 Capture Input	
		EC11	I	Timer 11 External Clock Input	
62	46	PB9*	IOUDS	PORT B Bit 9 Input/Output	Pull-up
		SPI21_SS	I/O	SPI Channel 21 Slave Select Signal	
		I2C0_SDA	I/O	I2C Channel 0 SDA In/Output	
		T12OUT	O	Timer 12 Output	
		T12CAP	I	Timer 12 Capture Input	
		EC12	I	Timer 12 External Clock Input	
63	47	VSS	P	VSS	
64	48	VDD	P	VDD	

NOTES:

1. I=Input, O=Output, U=Pull-up, D=Pull-down, S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power
2. The * means 'selected pin function after reset condition'.
3. Pin order may be changed with revision notice.
4. BOOT0, nRESET, PA13 (SWDIO), PA14 (SWCLK), PB8, and PB9 are the default pull-up pins.

3. System and memory overview

3.1 System architecture

Main system of A31G32x series consists of the followings:

- ARM® Cortex® -M0+ core
- General purpose DMA
- Internal SRAM
- Internal Flash memory
- Two AHB buses

3.1.1 Cortex-M0+ Core

The ARM® Cortex®-M0+ processor is the most energy-efficient ARM processor available. It builds on the very successful Cortex-M0+ processor, retaining full instruction set and tool compatibility, while further reducing energy consumption and increasing performance. Document "DDI 0484C" from ARM provides detail information of Cortex-M0+.

3.1.2 Interrupt controller

Table 3. Interrupt Vector Map

Priority	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Handler
-13	0x0000_000C	Hard Fault Handler
-12	0x0000_0010	Reserved
-11	0x0000_0014	
-10	0x0000_0018	
-9	0x0000_001C	
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Handler
-4	0x0000_0030	Reserved
-3	0x0000_0034	
-2	0x0000_0038	PenSV Handler
-1	0x0000_003C	SysTick Handler

Table 3. Interrupt Vector Map (continued)

Priority	Vector Address	Interrupt Source
0	0x0000_0040	LVR
1	0x0000_0044	SYSCLKFAIL
2	0x0000_0048	WDT
3	0x0000_004C	GPIOA, GPIOB
4	0x0000_0050	GPIOC, GPIOF
5	0x0000_0054	Reserved
6	0x0000_0058	Reserved
7	0x0000_005C	TIMER10
8	0x0000_0060	TIMER11
9	0x0000_0064	TIMER12
10	0x0000_0068	I2C0
11	0x0000_006C	USART10
12	0x0000_0070	WT
13	0x0000_0074	TIMER30
14	0x0000_0078	I2C1
15	0x0000_007C	TIMER20
16	0x0000_0080	TIMER21
17	0x0000_0084	USART11
18	0x0000_0088	ADC
19	0x0000_008C	SPI20
20	0x0000_0090	SPI21
21	0x0000_0094	TIMER13
22	0x0000_0098	TIMER40
23	0x0000_009C	RTC
24	0x0000_00A0	Reserved
25	0x0000_00A4	Reserved
26	0x0000_00A8	USART12
27	0x0000_00AC	USART13
28	0x0000_00B0	TEMP-Sensor
29	0x0000_00B4	COMP
30	0x0000_00B8	USB
31	0x0000_00BC	CRC

NOTES:

1. Each interrupt has an associated priority-level register. Each of them is 2 bits wide, occupying the two MSBs of the Interrupt Priority Level Registers.

Each Interrupt Priority Level Register occupies 1 byte (8 bits).NVIC registers in the Cortex-M0+ processor can only be accessed using word-size transfers, so for each access, four Interrupt Priority Level Registers are accessed at the same time.
** __NVIC_PRIO_BITS = 2

2. Figure 5 is a caution when using Peripheral Interrupts. Interrupt don't work if only peripheral interrupts are enabled. Enable the function in core to enable interrupt.
   * __enable_irq > NVCI_EnableIRQ(Peripheral) > Each Peripheral Interrupt

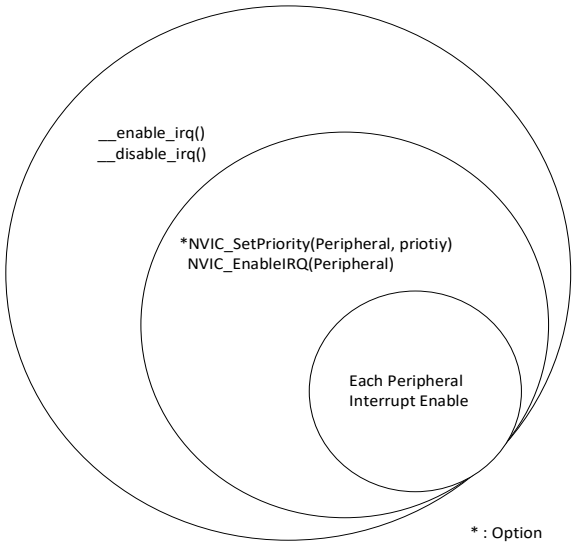


Figure 5. Interrupt Block Diagram

3.2 Memory organization

Program memory, data memory, registers and I/O ports are organized in the same address space.

3.2.1 Register boundary address

Table 4 gives the boundary addresses of peripherals in A31G32x series.

Table 4. A31G32x Memory Boundary Addresses

Boundary address	Memory area
0x4000_0000	SCU
0x4000_5100	LVI/LVR
0x4000_1000/1100/1200/1500	PCU A/B/C/F
0x4000_0100	Flash controller
0x2000_0000	Internal SRAM
0x4000_0400/0410/0420/0430	DMACH0/1/2/3
0x4000_6200	Static memory controller
0x4000_1A00	WDT
0x4000_2000	WT
0x4000_2100/2200/2300/2700	Timer 10/11/12/13
0x4000_2500/2600	Timer 20/21
0x4000_2400/2800	Timer 30/40
0x4000_3800/3900/3A00/3B00	USART 10/11/12/13
0x4000_4800/4900	I2C 0/1
0x4000_4C00/4D00	SPI 20/21
0x4000_3000	12-bit ADC
0x4000_3500	10-bit DAC
0x4000_3420	Comparator
0x4000_0300	CRC
0x5000_0000	USB
0x4000_6100	RTC
0x4000_6300	Temp sensor

3.2.2 Memory map

Figure 6 shows addressable memory space in memory map.

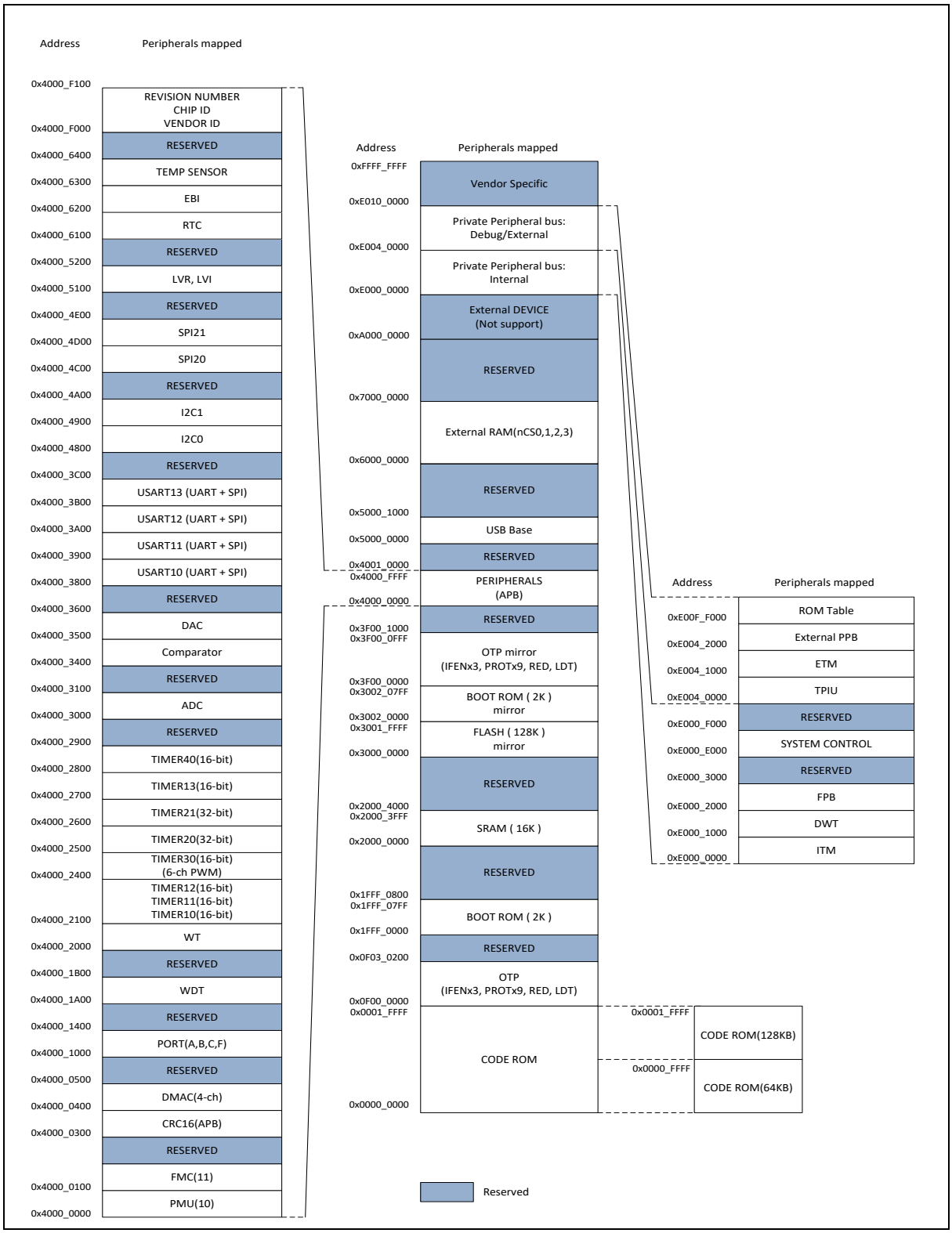


Figure 6. Memory Map

3.2.3 Embedded SRAM

A31G32x series have a block of 0-wait on-chip SRAM. The size of SRAM is 16KB and its base address is 0x2000_0000.

SRAM memory area is usually used for data memory and stack memory. Sometimes the code is dumped into the SRAM memory for fast operation or flash erase/programming operation. This device does not support memory remap strategy. So jump and return are required to perform the code in SRAM memory area.

3.2.4 Flash memory overview

A31G32x series provides internal 128KB code flash memory and its controller. This is enough to control the general system. Self-programming is available and ISP and SWD programming is also supported in boot or debugging mode.

Instruction and data cache buffer are present and overcome the low bandwidth flash memory. CPU can access flash memory with one wait state up to 48MHz bus frequency.

3.2.5 Boot mode

Boot mode pins

A31G32x series has a boot mode option to program internal flash memory. Boot mode can be entered by setting BOOT pin to 'L' at reset timing (Normal state is 'H').

Boot mode supports both of UART boot and I2C boot:

- UART boot uses USART10_TXD/USART10_RXD port or USART11_TXD/USART11_RXD.
- I2C boot uses I2C0_SCL/I2C0_SDA1 port.

The pins for boot mode are listed in Table 5.

Table 5. Boot Mode Pin List

Block	Pin Name	Dir	Description
SYSTEM	nRESET	I	Reset Input signal
	nBOOT	I	'Low' to enter Boot mode
UART mode of USART10	USART10_RXD/PA10	I	USART10 Boot Receive Data
	USART10_TXD/PA9	O	USART10 Boot Transmit Data
UART mode of USART11	USART11_RXD/PA15	I	USART11 Boot Receive Data
	USART11_TXD/PA14	O	USART11 Boot Transmit Data
I2C	I2C0_SDA/PB7	I / O	I2C0 Boot Data Input/Output
	I2C0_SCL/PB6	I	I2C0 Boot Clock Input

Boot mode connections

User can design a target board using any of boot mode ports such as I2C or UART mode of USART10, USART11. Sample connection diagrams of boot mode are introduced in the following figures:

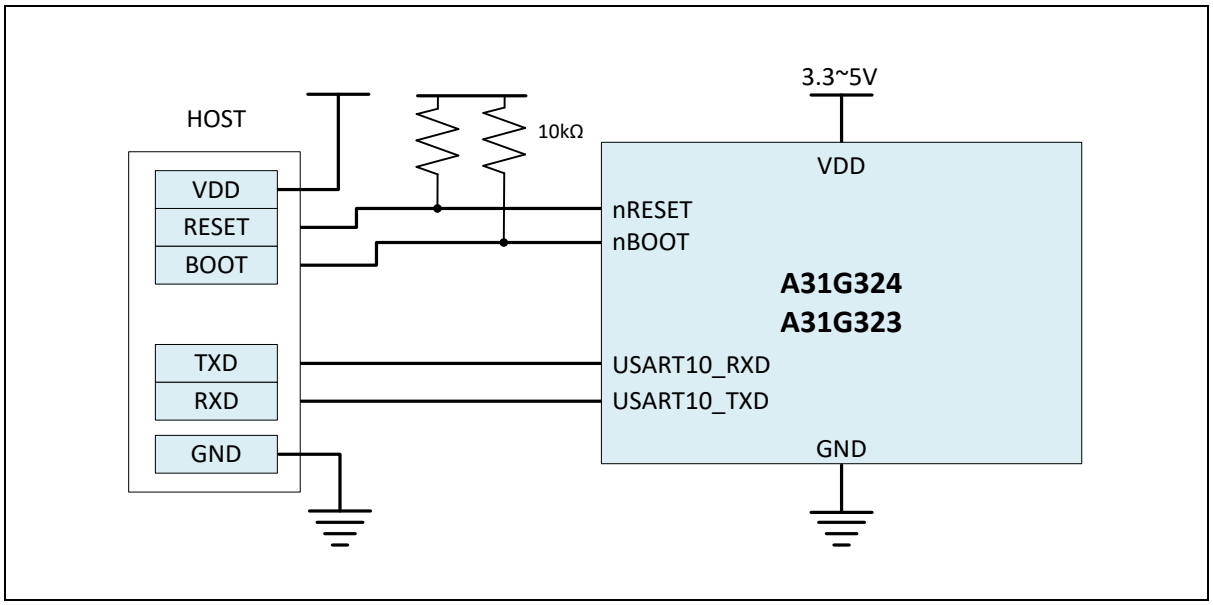


Figure 7. Connection Diagram of UART10 Boot

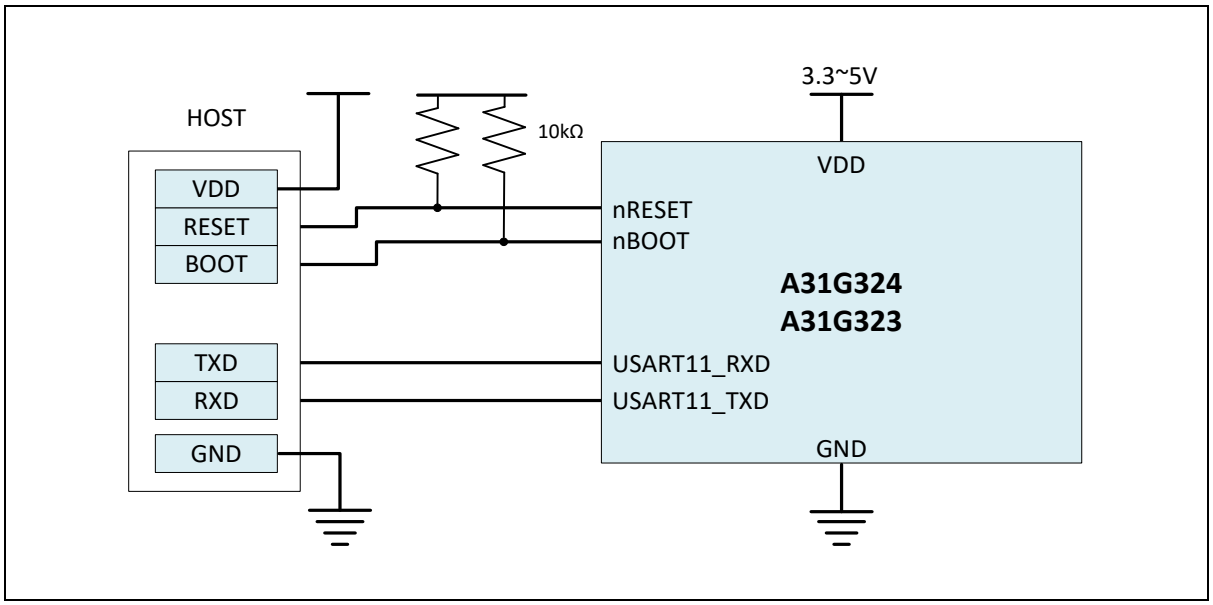


Figure 8. Connection Diagram of UART11 Boot

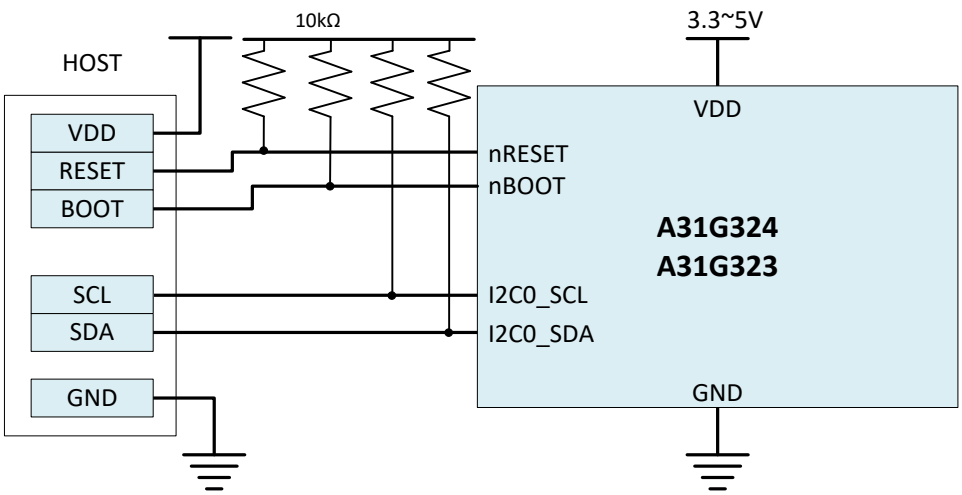
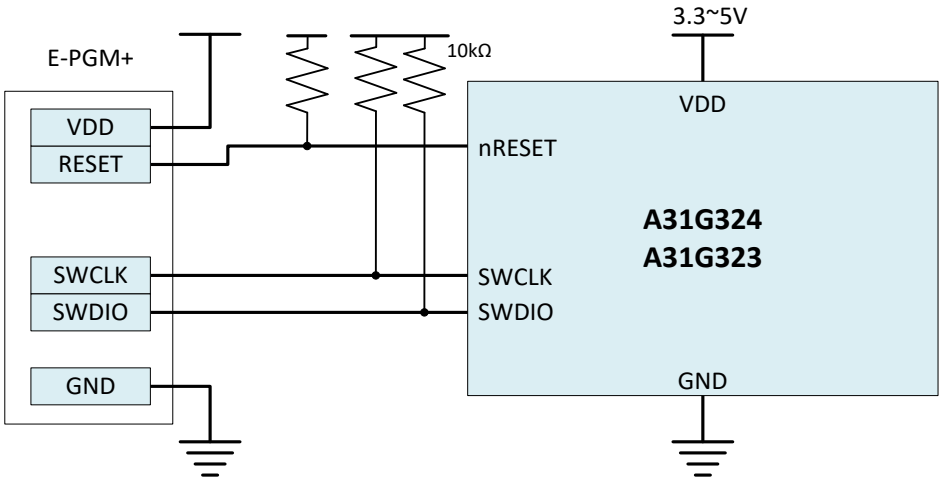


Figure 9. Connection Diagram of I2C Boot

SWD mode connections

A user can use SWD mode for writing with E-PGM+. This mode can be used for writing & debugging.



NOTES:

1. A31G32x series should not be connected to SWD mode in Boot Mode.
2. SWCLK Pin is used in Boot Mode as USART11_TXD.

Figure 10. Connection Diagram of E-PGM+ and SWD Port

4. System control unit

A31G32x series have a built-in intelligent power control block which manages system analog blocks and operating modes. System control unit (SCU) block controls an internal reset and clock signals to maintain optimize system performance and power dissipation.

Four pins in Table 6 are assigned for SCU block.

Table 6. SCU Pins

Pin name	Type	Description
nRESET	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator
SXIN/SXOUT	OSC	External sub-Crystal Oscillator
CLKO	O	Clock Output Monitoring Signal

4.1 SCU block diagram

In this subsection, SCU block diagram is introduced in Figure 11.

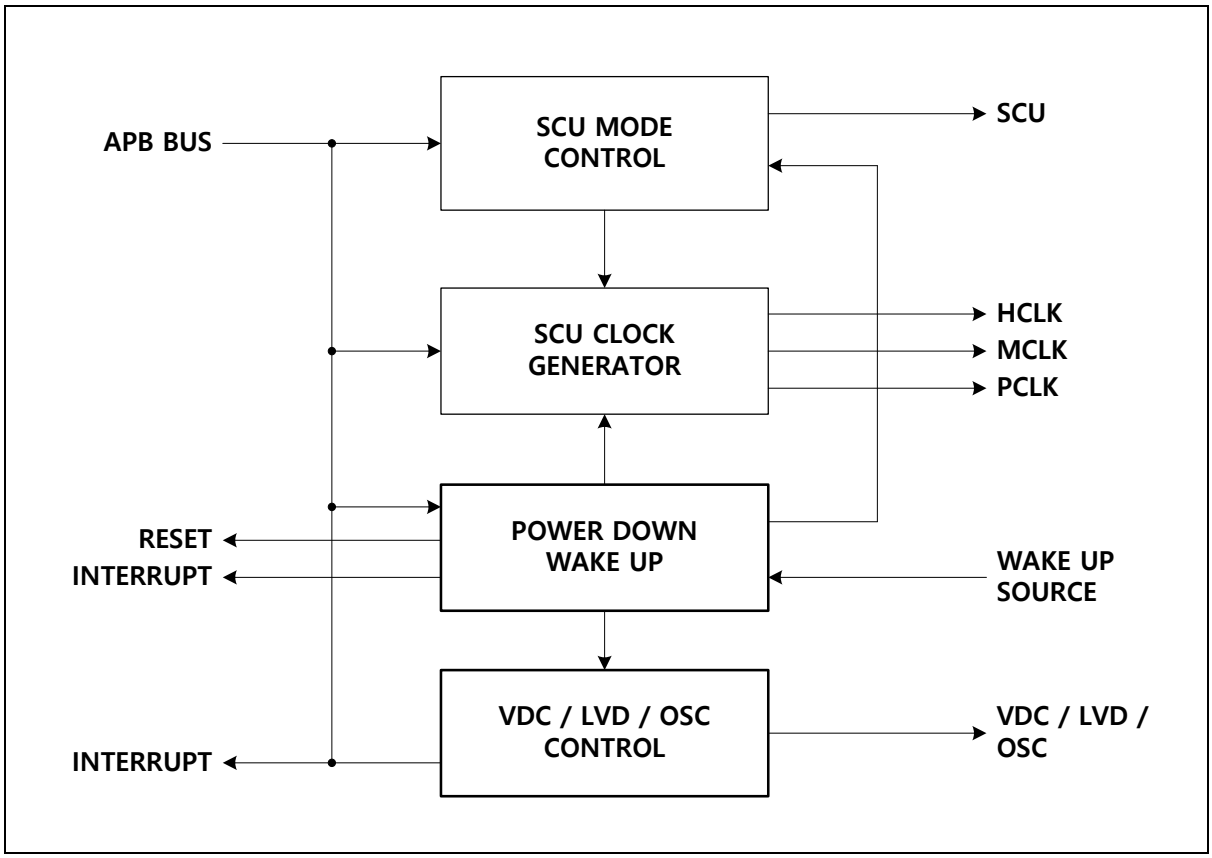


Figure 11. SCU Block diagram

4.2 Clock system

A31G32x series have two main operating clocks. One is HCLK which produces a clock signal both for CPU and AHB bus system. The other is PCLK which produces a clock signal for peripheral systems.

A user can keep the clock system variation under software control. Through Figure 12 and Table 7, users learn about the clock system of A31G32x devices and clock sources.

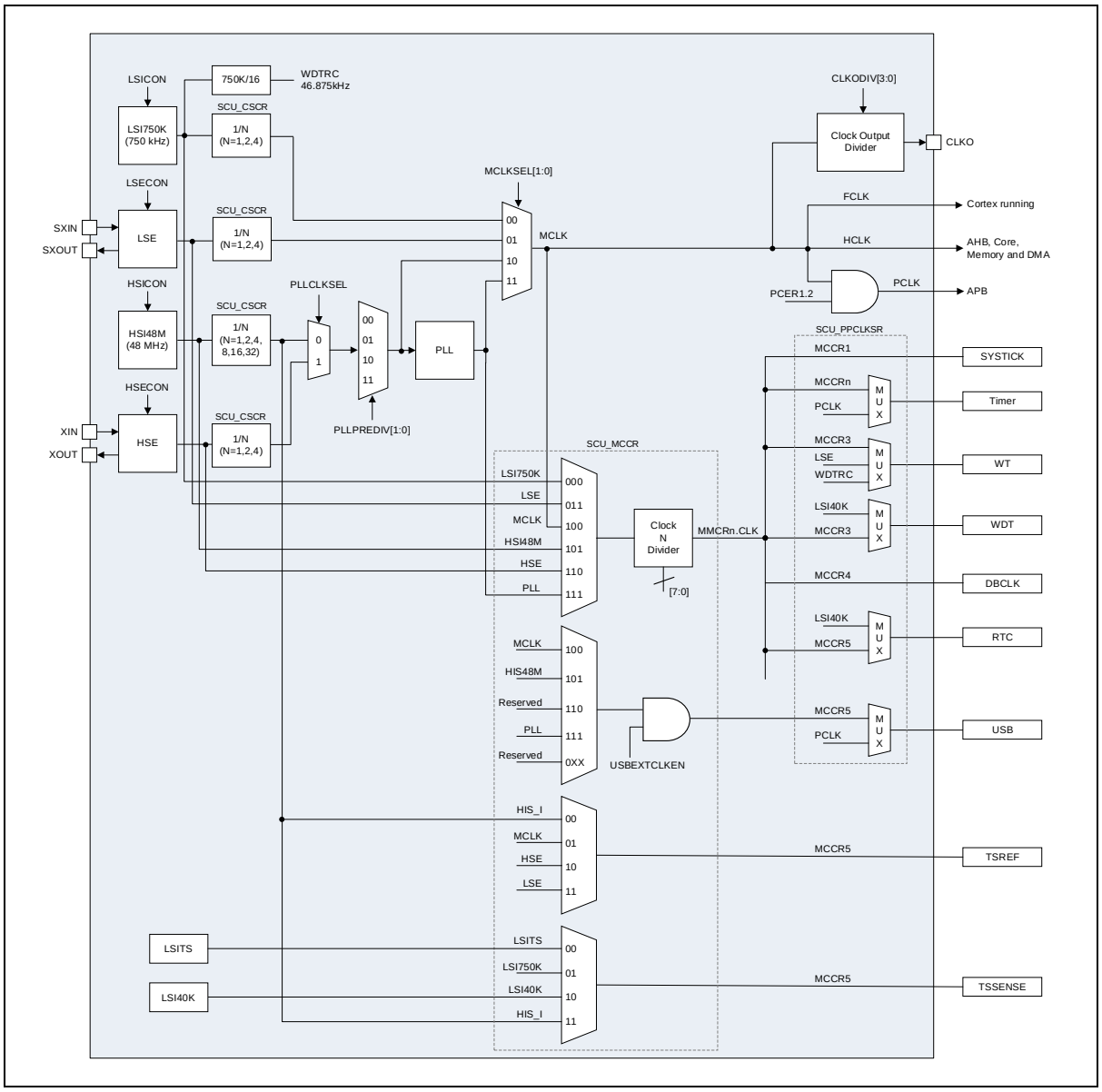


Figure 12. Clock Tree Configuration

All multiplexers switching clock sources have a glitch-free circuit in each. So a clock can be switched without glitch risks. When a user tries to change a clock mux control, both of clock sources must be alive. If one of them is not alive, clock change operation is stopped and system will be halted and not be recovered.

Table 7. Clock Sources

Clock name	Frequency	Description
HSE	2-16 MHz	High Speed External Oscillator
LSE	32.768 kHz	Low Speed External Oscillator
HSI	48 MHz	High Speed Internal OSC
LSI750K	750 kHz	Low Speed Internal OSC
LSI40K	40 kHz	Low Speed Internal OSC
LSITS	—	Internal OSC for temp sensor

4.2.1 HCLK clock domain

HCLK clock feeds the clock to the CPU and the AHB bus. Cortex-M0+ CPU requires 2 clocks related with FCLK and HCLK. FCLK is free running clock and it is always running except in power down mode. HCLK can be stopped in SLEEP mode and power down mode.

BUS system and memory systems are operated by MCLK clock. Maximum bus operating clock speed is 48MHz.

4.2.2 PCLK clock domain

PCLK is the master clock of all peripherals. Each peripheral clock is enabled by SCU_PCER1, and SCU_PCER2 registers can be used by each peripheral. Before enabling the PCLK input clock of each block, it can't be accessible even reading its registers. It can be stopped in power down mode.

4.2.3 Clock configuration procedure

After powering up, the default system clock is fed by LSI750K (750KHz) clock. By default LSI750K is enabled at power up sequence. The other clock sources will be enabled by user controls with the LSI750K system clock.

HSI48M (48MHz) clock can be enabled by SCU_CSCR register.

HSE (2-16MHz) clock can be enabled by SCU_CSCR register. Prior to enable the HSE block, the pin mux configuration should be set for XIN, XOUT function. PF0 and PF1 pins are shared with HSE's XIN and XOUT function – PF_MOD and PF_AFSR0 registers should be configured properly. After enabling the HSE block, you must wait for more than 2ms time to ensure stable operation of crystal oscillation.

LSE (32.768KHz) clock can be enabled by SCU_CSCR register. Prior to enable the LSE block, the pin mux configuration should be set for SXIN, SXOUT function. PC14 and PC15 pins are shared with LSE's SXIN and SXOUT function – PC_MOD and PC_AFSR1 registers should be configured properly.

After enabling the LSE block, you must wait for more than 2s time to ensure stable operation of crystal oscillation. You can change an MCLK by using the SCU_SCCR register.

You can find an example flow chart configuring the system clock in Figure 13.

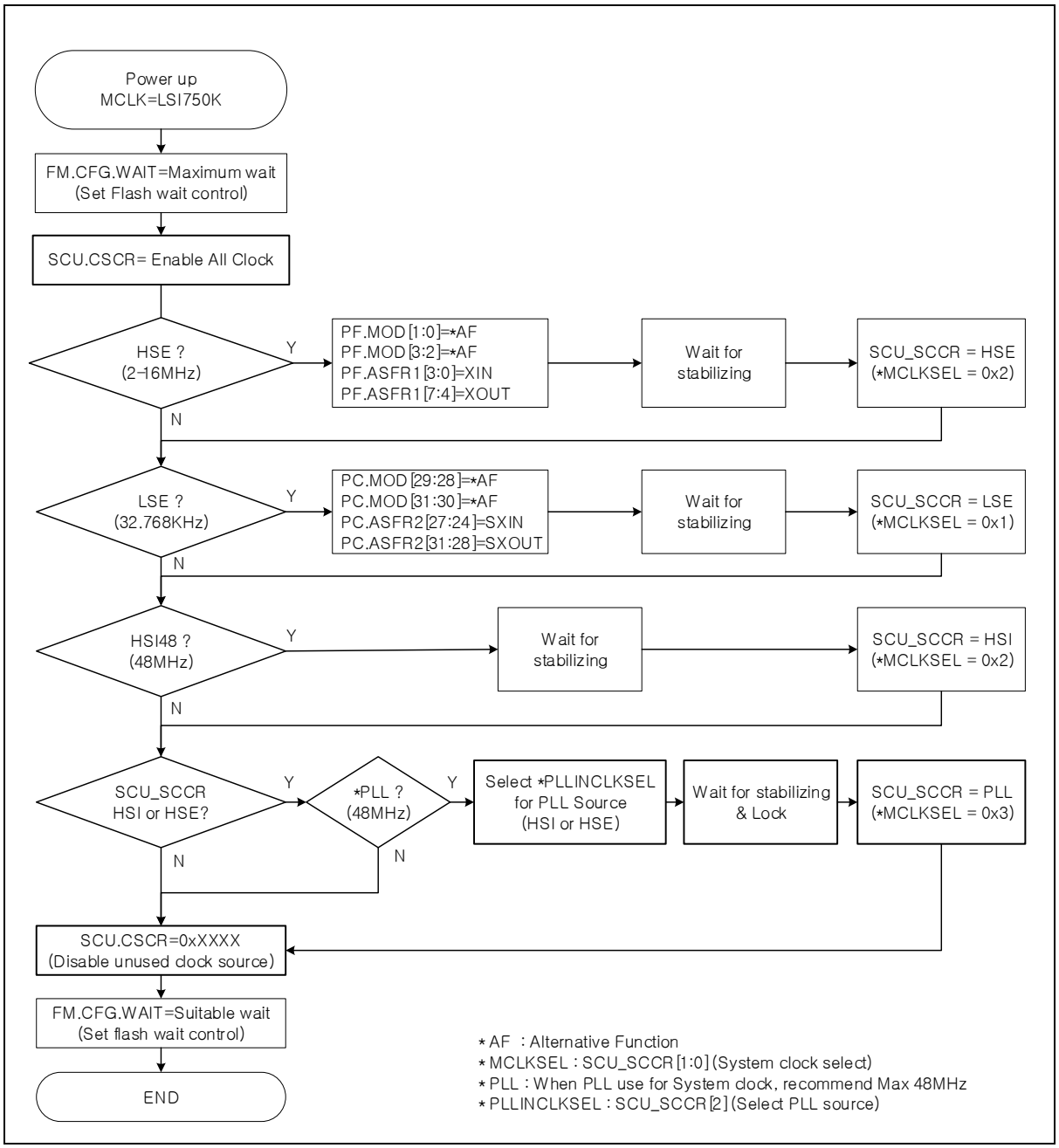


Figure 13. Clock Change Procedure

When you speed up the system clock to the maximum operating frequency, you should check flash wait control configuration. Flash read access time is one of limitation factor for performance. The wait control recommendation is suggested in Table 8.

Table 8. Flash Wait Control Recommendation

FM.CFG.WAIT	FLASH Access Wait	Available Max System clock frequency
00	0 clock wait	~20MHz
01	1 clock wait	~40MHz
10	2 clock wait	~48MHz

Figure 14 shows how to set the peripheral clock. Selecting a peripheral clock is a typical method of MCCRn and PCLK. Exceptionally WT, WDT, RTC use other clocks besides MCCRn and PCLK. (n = 1, 2, 3, 4, 5 and 6).

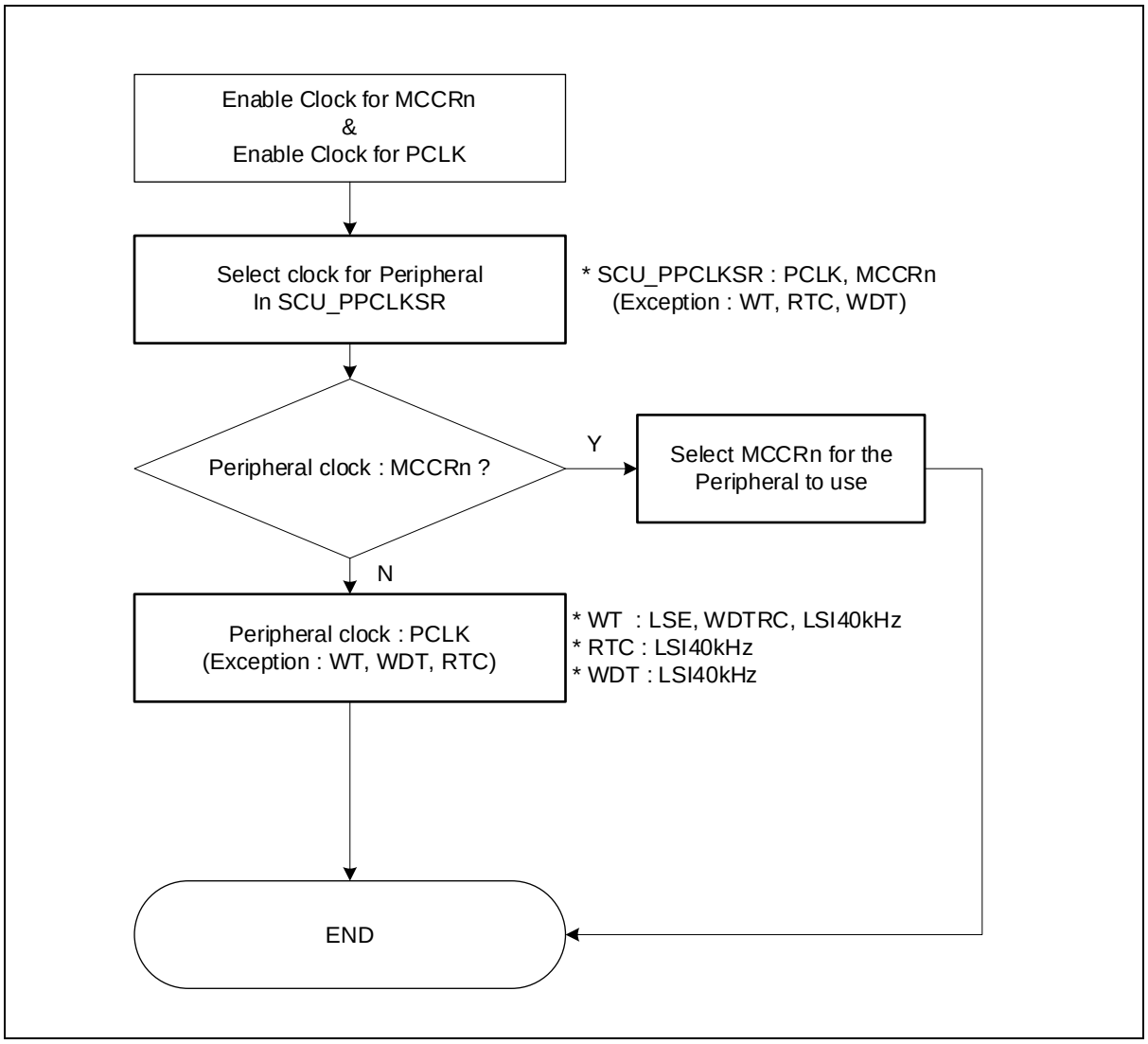


Figure 14. Peripheral Clock Select (n: 1, 2, 3, 4, 5, and 6)

Table 9. Peripheral Clock Select

Peripheral	Clock Selection		
	MCCRn	**PCLK**	**Another**
Systick	MCCR1	N/A	N/A
Timer1n		O	N/A
Timer20	MCCR2	O	N/A
Timer30		O	N/A
WT	MCCR3	N/A	WDTRC, LSE, LSI40kHz
WDT		N/A	LSI40kHz
DBCLK	MCCR4	N/A	N/A
RTC	MCCR5	N/A	LSI40kHz
TSREF		N/A	N/A
TSSENSE		N/A	N/A
USB		O	N/A
Timer40	MCCR6	O	N/A
Timer21		O	N/A

4.3 Power domain

A31G32x series have two VDCs supplying internal 1.5V power to digital circuit: one for VDD15 domain and one for backup domain.

VDD is the main power supply and VBAT is the backup power supply. A power supply switch determines which power is to be supplied (VPWRSW). In initial state, this switch selects VDD as power supply, so it isn't booted by VBAT but is booted only by VDD. To retain the contents of the backup domain when VDD is off during operation, the power supply switch selects VBAT and uses the optional STANDBY voltage provided by the battery or other source.

To improve conversion accuracy and to extend the supply flexibility, the ADC, the DAC and Comparator have an independent power supply VDDA and VSSA which can be separately filtered and shielded from noise on the PCB. The VDDA supply voltage must be greater or equal to VDD.

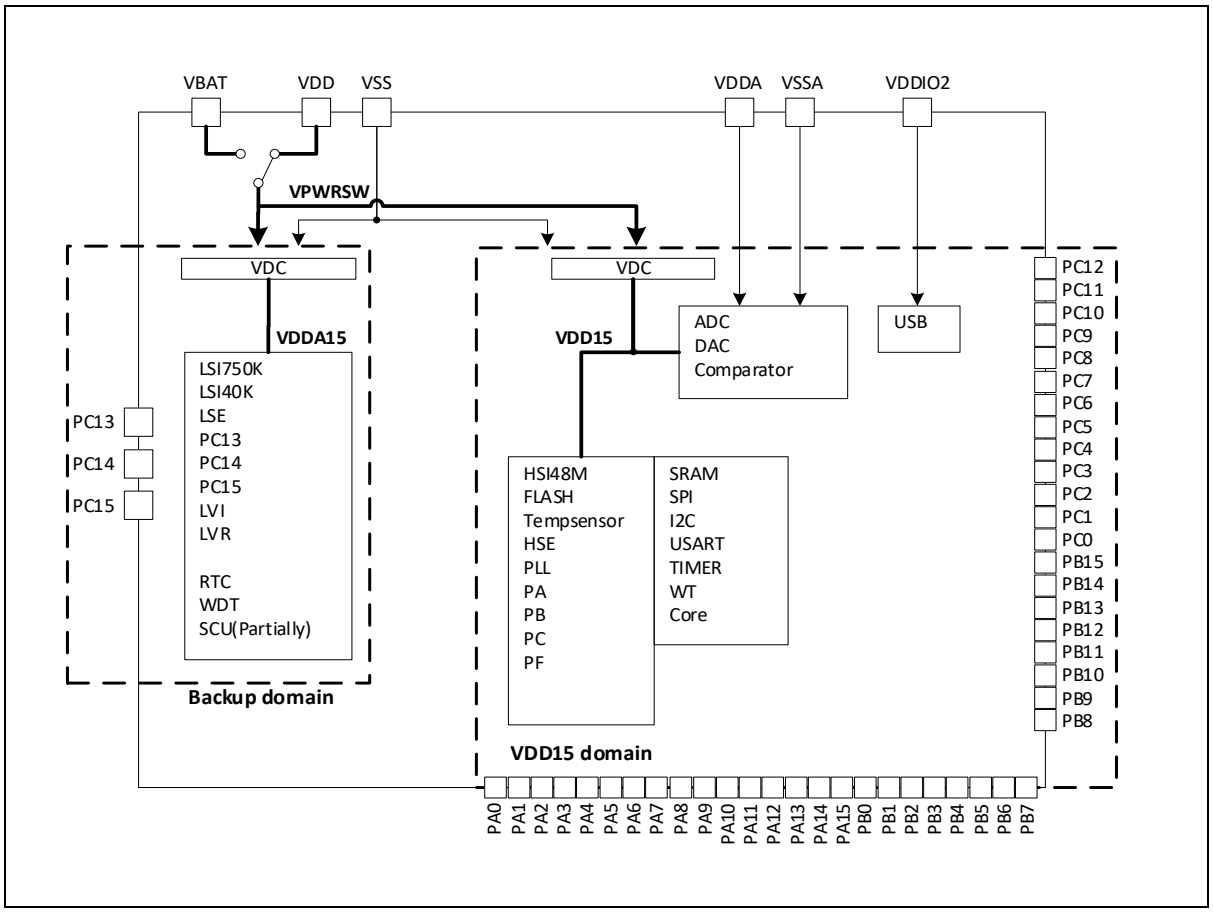


Figure 15. Power Domain Diagram

Another regulator VDC33 is embedded to supply 3.3V power to a USB. When power to the USB is supplied from outside, output of the VDC33 must be floating. VDD33_PD of SCU_VDCCON register must be set to 1 to make it floating.

4.4 Reset

A31G32x series have two system reset options. One is to cold reset that is effective during power up or down sequence. The other is warm reset which is generated by several reset sources. A reset event makes a chip to turn to an initial state. Reset sources of the cold reset and the warm reset are listed in Table 10.

Table 10. Reset Sources of Cold Reset and Warm Reset

	Cold reset	Warm reset
Reset sources	VPWRSW POR VDD15 POR LVR reset	nRESET Pin WDT reset MCLK Fail reset HSE Fail reset S/W reset CPU request reset

4.4.1 Cold reset

Cold reset is an important feature of a chip when power is up. This characteristic will affect overall system boot.

Internal VDC is enabled when VDD power is turn on. Internal POR trigger level is 1.2V of VDD voltage out level. At this time, boot operation is started. The LSI750K clock is enabled and counts 2.73msec time for internal VDC level stabilizing. In this time, VDD voltage level should be over than initial LVR level (1.63V). After 2.73msec counting, the cold reset is released and counts 0.27msec time for warm reset synchronizing. After releasing both cold and warm reset, BOOTROM and CPU are running.

Figure 16 shows power up sequence and internal reset waveforms.

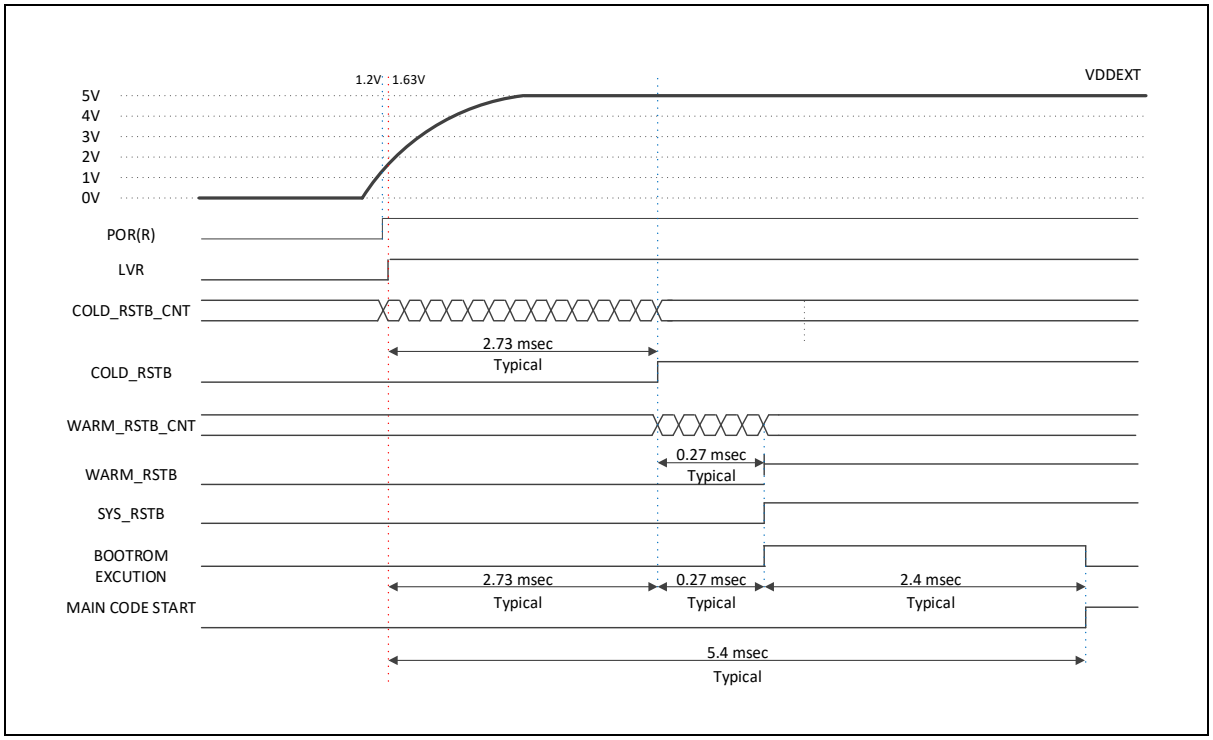


Figure 16. Power up Procedure

The VDD15 POR in the cold reset source senses the VDC output voltage in the VDD15 domain and generates a reset signal. The VDD15 POR reset signal initializes only the VDD15 domain. The backup domain is initialized only by VPWRSW POR or LVR reset.

In STANDBY mode and BACKUP POWER mode, VDD15 domain VDC is turned off and VDD15 POR occurs. Because core exists in VDD15 domain, wakeup in STANDBY mode or BACKUP POWER mode will execute BOOTROM and jump to start address.

4.4.2 Warm reset

Warm reset event has several reset sources and some parts of chip returns to an initial state when the warm reset condition is occurred.

The warm reset source is controlled by SCU_RSER register and the status is appeared in SCU_RSSR register. The reset for each peripheral blocks is controlled by SCU_PRER register. The reset can be masked independently.

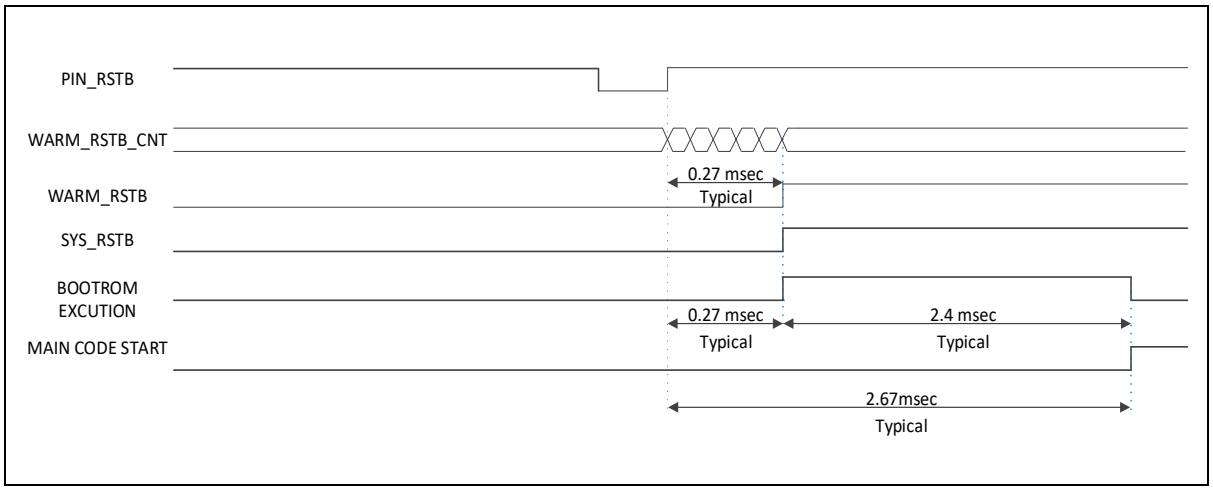


Figure 17. Warm Reset Diagram

4.4.3 LVR reset

Voltage level of LVR is set by low voltage reset configuration register (SCULV_LVRCNFG). Reset status of the LVR is shown in SCU_RSSR register. The LVR reset is controlled by SCULV_LVRCR register, which is cleared to "0x00" by VPWRSW POR reset.

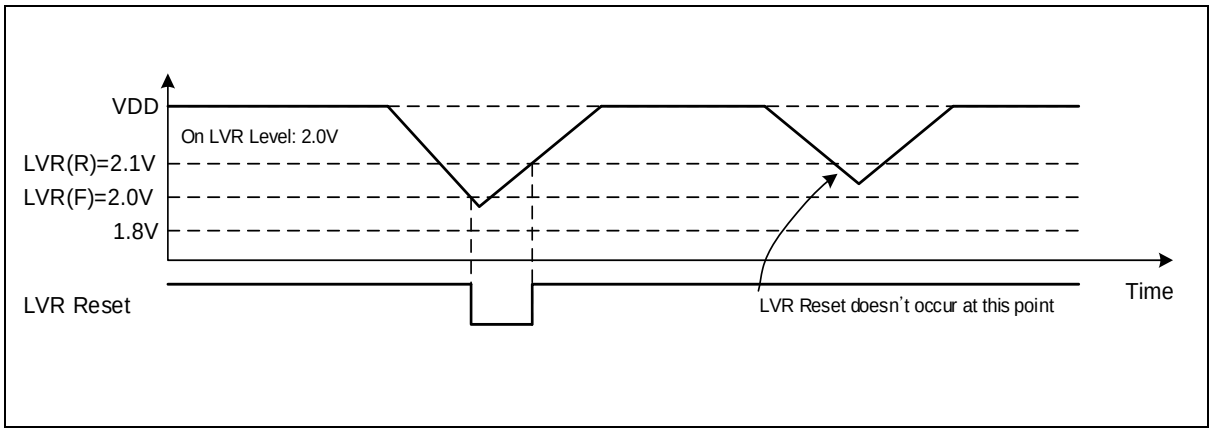
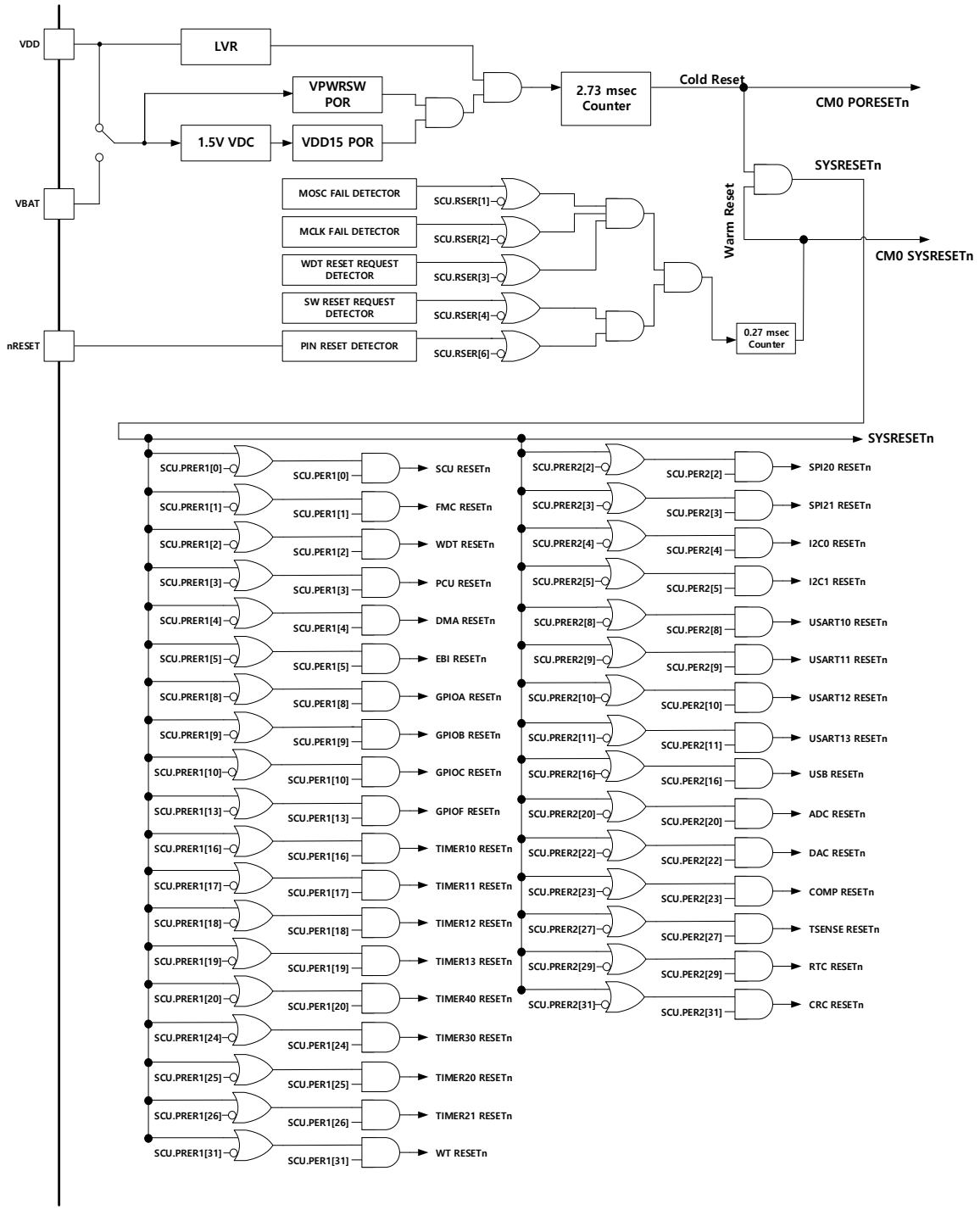


Figure 18. LVR Reset Timing Diagram

4.4.4 Reset tree



NOTES:

1. When cold reset occurs, WDT and RTC are no reset in STANDBY mode or BACKUP POWER mode.
2. WDT and RTC are reset, only by LVR reset and VPWRSW POR.

Figure 19. Reset Tree Configuration

4.5 Operation mode

INIT mode is an initial state of the chip when a reset is asserted. In RUN mode, CPU shows the maximum performance with high-speed clock system. SLEEP mode and three Power Down modes (STOP, STANDBY, and BACKUP POWER) can be used as a low power consumption mode. In the low power consumption mode, power is effectively managed to reduce power consumption by halting processor core and unused peripherals. Figure 20 describes transition between the operation modes.

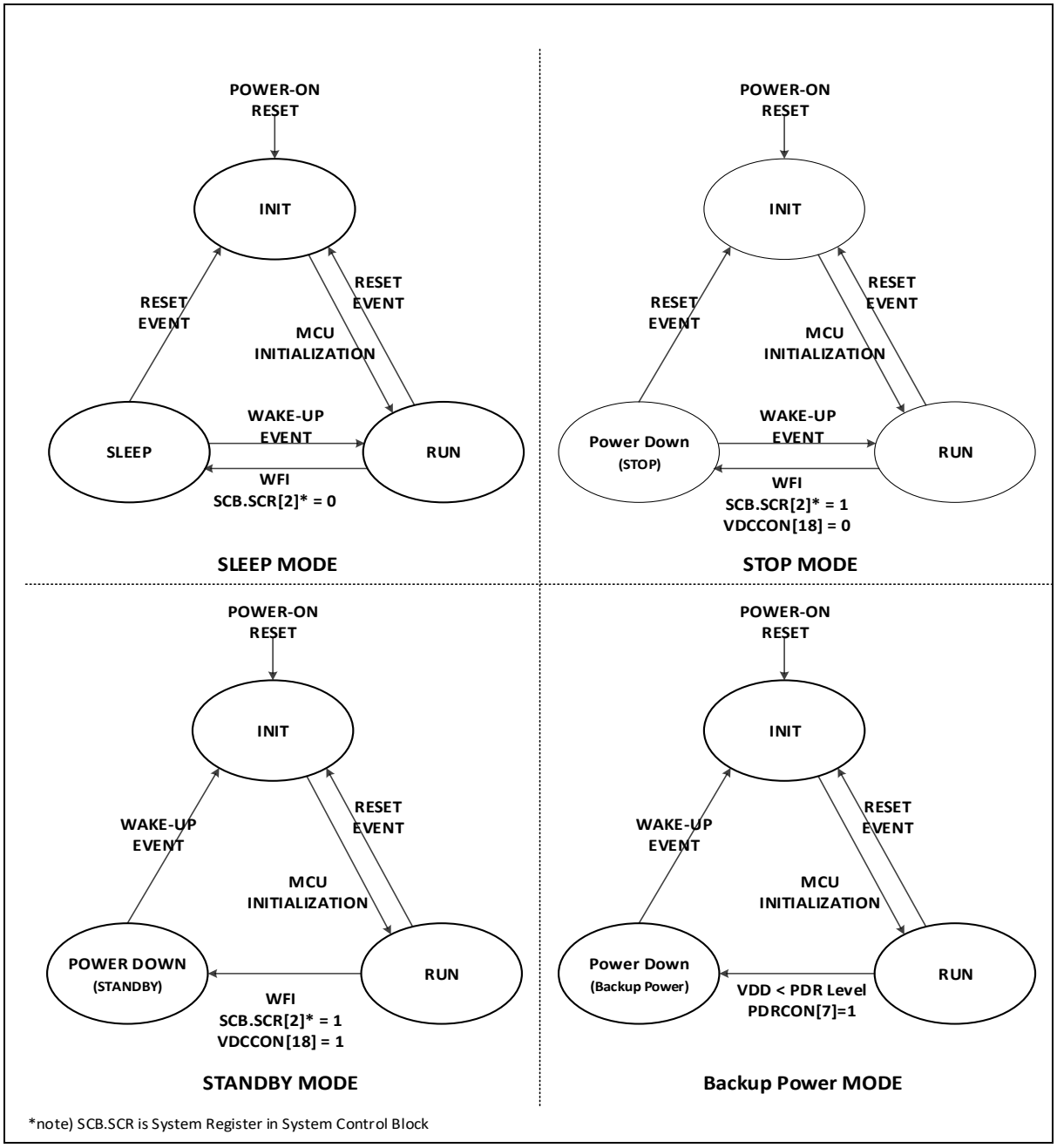


Figure 20. Transition between Operation Modes

Table 11. Operation Mode

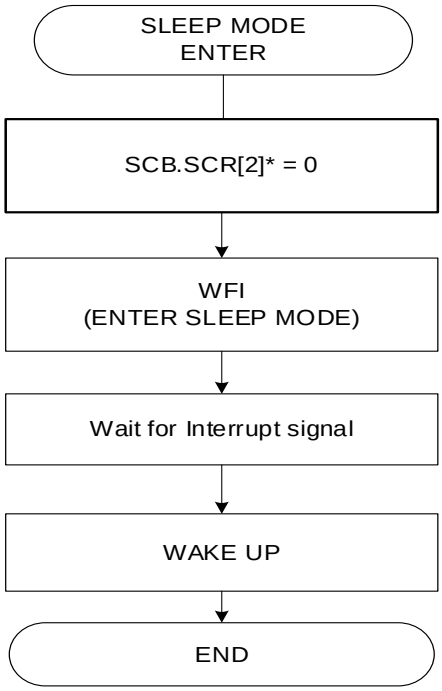
MODE	Condition	After Wake up Event	After Reset Event
RUN	POWER ON	N/A	INIT
SLEEP	WFI (Wait for Interrupt): SCB.SCR[2]*=0	RUN	INIT
STOP	WFI (Wait for Interrupt): SCB.SCR[2]*=1	RUN	INIT
STANDBY	WFI (Wait for Interrupt): SCB.SCR[2]*=1, VDDCON[18]=1	INIT	INIT
BACKUP POWER	VDD<PDR Level: PDRCON[7]=1	INIT	INIT

4.5.1 RUN mode

In RUM mode, CPU and the peripheral hardware operate with a high-speed clock. After a reset followed by INIT state, the system enters in the RUN mode.

4.5.2 SLEEP mode

Only CPU is stopped in this mode. Each peripheral function can be enabled by the function enable and clock enable bit in the PER and PCER register.



NOTE: SCB.SCR is System Control Register in System Control Block

Figure 21. SLEEP Mode Operation Sequence

4.5.3 Power-down mode

The A31G32x series have three power down modes: STOP mode, STANDBY mode, and BACKUP POWER mode.

In STOP mode, 1.5V VDC is off and backup domain VDC is supplied to 1.5V domain. All peripherals in the 1.5V domain are switched to STOP state and CPU goes under deepsleep state. In STANDBY mode, 1.5V VDC is off and power is not supplied to 1.5V domain. Therefore, all peripherals of the 1.5V domain are initialized by VDD15 POR. In BACKUP POWER mode, VDD is turned off and the power switch selects VBAT instead of VDD. 1.5V VDC is off and no power is supplied and initialized by VDD15 POR.

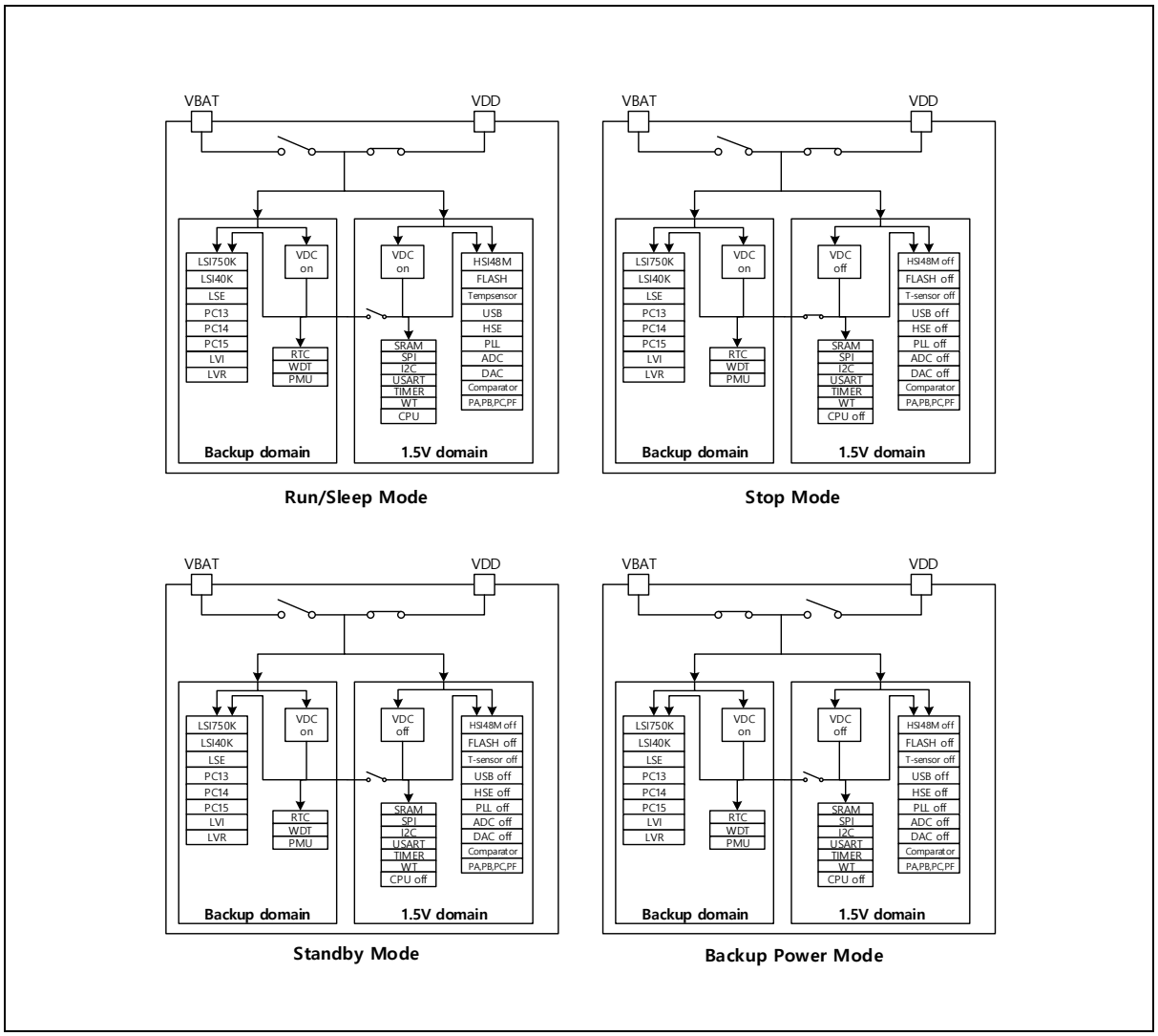


Figure 22. Power down Mode Block Diagram

LVI operates in STOP mode and STANDBY mode but is ignored in BACKUP POWER mode. LVR operation is not recommended in STOP mode and STANDBY mode. To use it, set BGRAON = 1 in SCU_SMR. LVR is not supported in BACKUP POWER mode. LVI and LVR are set using the SCULV_LVRCNFIG, SCULV_LVRCR and SCULV_LVICR registers. Initially the LVR is on. When LVR occurs, both backup domain and 1.5V domain are reset.

In power down mode, LSE, and LSI40K can operate in backup domain. To operate the LSE in power down mode, set SCU_CSCR [14] = 1. LSI40K does not turn off automatically in power down mode, so if you want LSI40K not to work in power down mode, you have to turn it off before entering power down mode.

Table 12. Oscillator Control in Power down Mode

	Operation enable	Power down operation
LSE	SCU_CSCR[15] = 1	If SCU_CSCR[14] = 1, RUN mode state is maintained If SCU_CSCR[14] = 0, turned off.
LSI40K	SCU_MCCR3[11] = 1	RUN mode state is maintained.

Table 13 lists peripheral's operation in each power down mode.

Table 13. Peripheral Operation in Power down Mode

Mode	Stop mode	Standby mode	Backup power mode
HSI48M	Stop	Power off	Power off
FLASH	Stop	Power off	Power off
Tempsensor	Stop	Power off	Power off
USB	Stop	Power off	Power off
HSE	Stop	Power off	Power off
PLL	Stop	Power off	Power off
ADC	Stop	Power off	Power off
DAC	Stop	Power off	Power off
Comparator	Operates	Power off	Power off
CPU	Deepsleep	Power off	Power off
SRAM	Retain	Power off	Power off
SPI20,21	Stop	Power off	Power off
I2C0,1	Operates only slave mode	Power off	Power off
USART10,11,12,13	Operates as only wakeup source	Power off	Power off
TIMER10,11,12,13 TIMER20,21,30,40	Operates only when the TIMER clock is oscillates	Power off	Power off

Table 13. Peripheral Operation in Power down Mode (continued)

Mode	Stop mode	Standby mode	Backup power mode
WT	Operates only when the WT clock is oscillates	Power off	Power off
PA,PB,PC,PF (except PC13, PC14, PC15)	Retain	Power off	Power off
PC13	Retain	GPIO or wakeup Pin	GPIO or wakeup Pin
PC14	Retain	GPIO or SXIN	GPIO or SXIN
PC15	Retain	GPIO or SXOUT	GPIO or SXOUT
LSI750K	Off	Off	Off
LSI40K	Operates	Operates	Operates
LSE	Operates	Operates	Operates
LVI	Operates	Operates	Discard
LVR	Operates	Operates	No support
RTC	Operates only when the RTC clock is oscillates	Operates only when the RTC clock is oscillates	Operates only when the RTC clock is oscillates
WDT	Operates only when the WDT clock is oscillates	Operates only when the WDT clock is oscillates	Operates only when the WDT clock is oscillates
SCU(Partially)	Operates	Operates	Operates

4.5.4 STOP mode

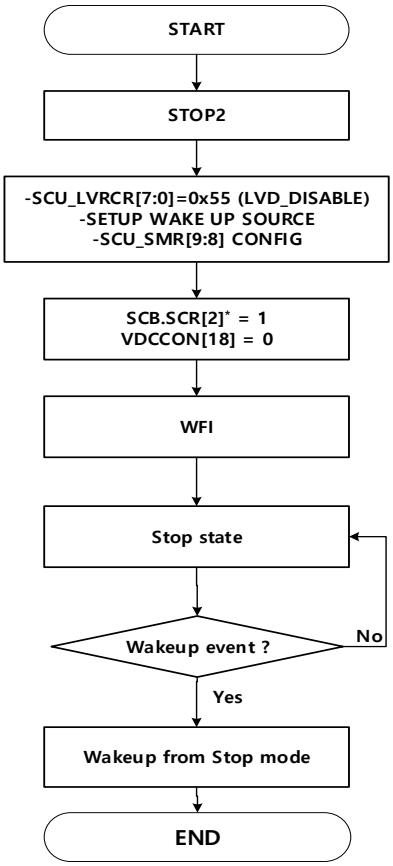
When the core goes under deepsleep state using WFI instruction, the chip enters in STOP mode. In STOP mode, all peripherals of 1.5V domain are stopped. Once power is supplied, internal SRAM and registers maintain their values.

The backup domain operates regardless of the STOP mode. To wake up in STOP mode, an interrupt request must be generated and can be selected in the SCU_WUER register. Stabilization time is 2.73ms after wakeup event occurs.

When entering in STOP mode, HSE is selected as the PLL input clock. These are maintained after wakeup.

Table 14. STOP Mode Configuration

Mode	Condition	Wakeup source
STOP	WFI SCB.SCR[2]* = 1 VDCCON[18] = 0	Source included in WUER



NOTES:

1. SCB.SCR is System Control Register in System Control Block.

Figure 23. STOP Mode Sequence

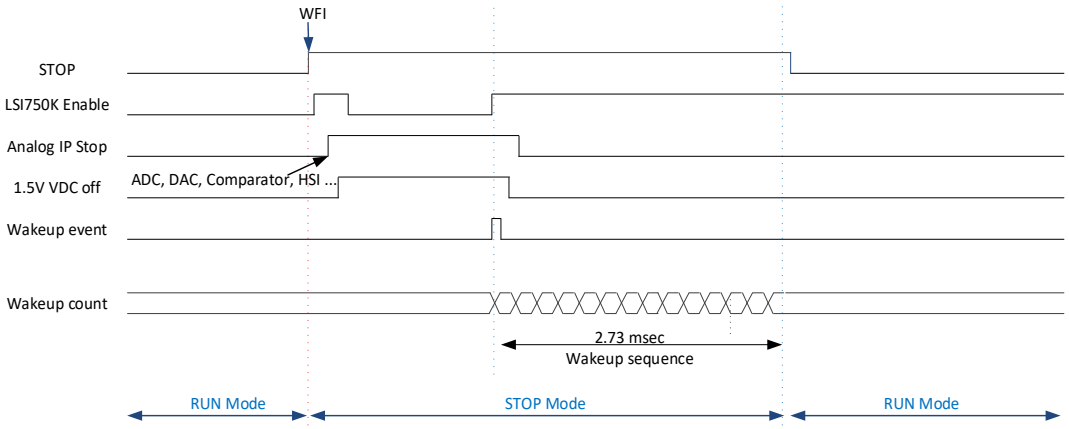


Figure 24. STOP Mode Timing Diagram

4.5.5 STANDBY mode

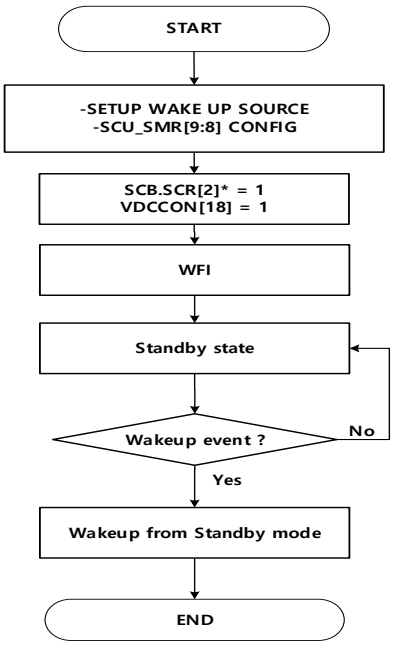
When the core goes under deepsleep state using WFI instruction with VDCCON[18] = 1, the chip enters in STANDBY mode. In STANDBY mode, VDC is turned off in 1.5V domain and all peripherals of 1.5V domain are in power off state. Therefore, the 1.5V domain is initialized as a whole by the VDD15 POR in STANDBY mode. The backup domain operates regardless of the STANDBY mode.

When selected in the SCU_WUER register, the LVI, WDT, and RTC interrupt flags in the backup domain can be used to wake up in STANDBY mode. If you set PC13 as wakeup pin by setting the RTCPFCR register, the chip can wake up from STANDBY mode with rising edge of PC13. When a pin reset occurs in STANDBY mode, the chip exits STANDBY mode. Pin reset can be enabled in SCU_RSER. Stabilization time is 2.73ms after wakeup event occurs.

When entering in STANDBY mode, HSE is selected as the PLL input clock. These are maintained after wake-up.

Table 15. STANDBY Mode Configuration

Mode	Condition	Wakeup source
STANDBY	WFI SCB.SCR[2]* = 1 VDCCON[18] = 1	RTC WDT PC13 (Wakeup pin) Pin reset



NOTES:

1. SCB.SCR is System Control Register in System Control Block

Figure 25. STANDBY Mode Sequence

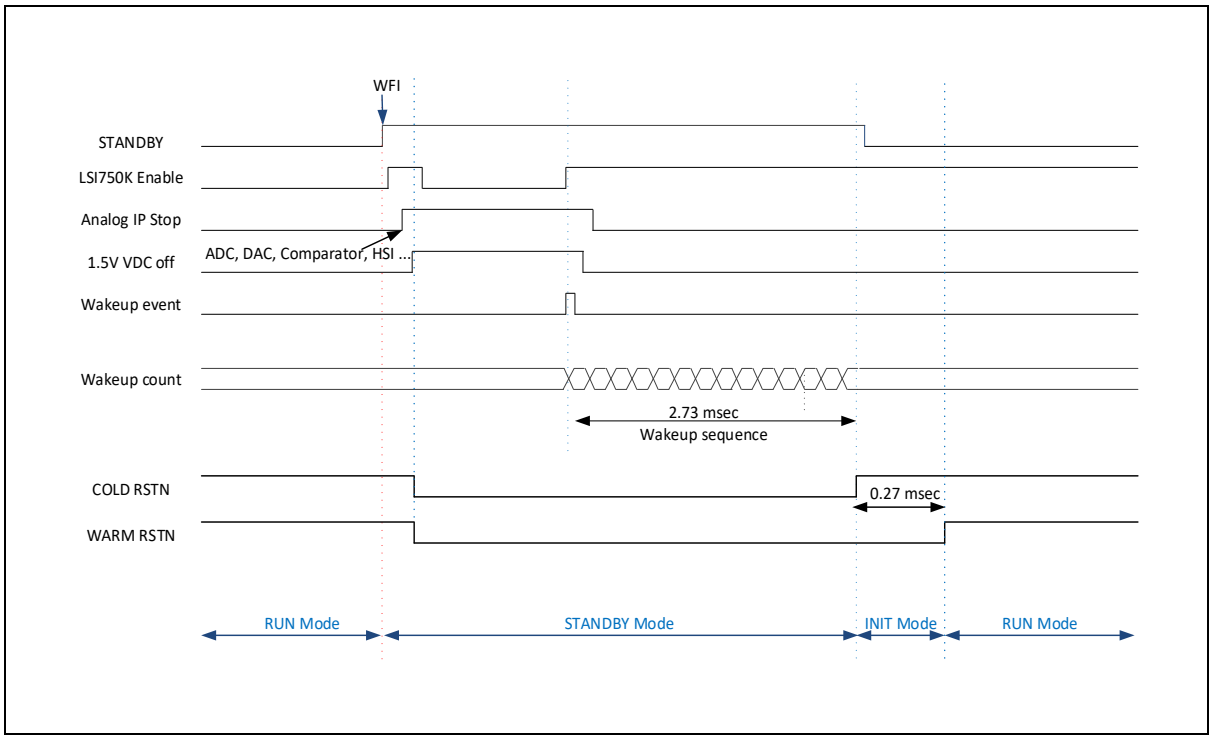


Figure 26. STANDBY Mode Timing Diagram

4.5.6 BACKUP POWER mode

By connecting the VBAT pin to an optional standby voltage from a battery or other source, you can maintain the contents of the backup register and power the entire backup domain when VDD is turned off.

BACKUP POWER mode can be entered from all operation modes (RUN, SLEEP and STOP, STANDBY). If a PDR reset occurs when Power down reset (PDR) is enabled through the PDRCON register, the chip enters in BACKUP POWER mode. In BACKUP POWER mode, 1.5V VDC is turned off and all peripherals are in power off state. If VDD rises again and becomes higher than PDR, it exits BACKUP POWER mode and 1.5V domain restarts in initial state. For more information about BACKUP POWER mode, please refer to 4.3 Power domain.

Table 16. BACKUP POWER Mode Configuration

Mode	Condition	Wakeup source
BACKUP POWER	PDRCON[7] = 1	(VDD rising) > PDR

NOTE: It is recommended that VBAT is higher than PDR voltage.

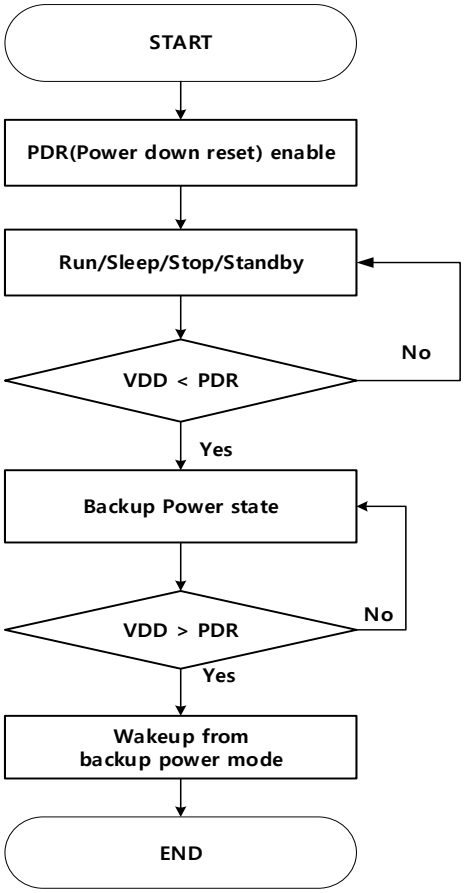


Figure 27. BACKUP POWER Mode Sequence

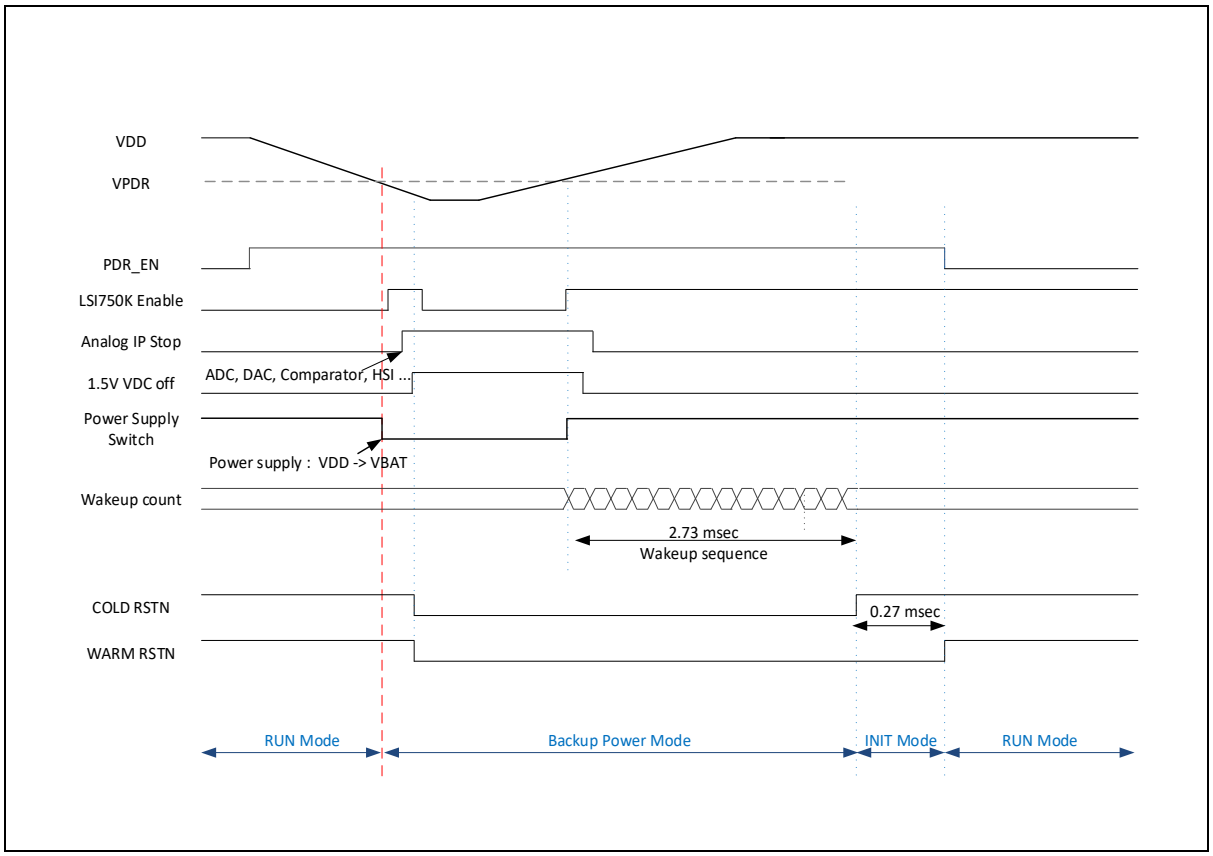


Figure 28. BACKUP POWER Mode Timing Diagram

4.6 Registers

Base address of SCU (chip configuration) and register map are introduced in the followings:

Table 17. Base Address of SCU (Chip Configuration)

Name	Base address
SCU (Chip Configuration)	0x4000_F000

Table 18. SCU Register Map (Chip Configuration)

Name	Offset	Type	Description	Reset value	Reference
SCUCC_VENDORID	0x0000	RO[NOTE]	Vendor Identification Register	0x4142_4F56	4.6.1
SCUCC_CHIPID	0x0004	RO[NOTE]	Chip Identification Register.	0x4D31_A00x	4.6.2
SCUCC_REVNR	0x0008	RO[NOTE]	Revision Number Register	0x0000_00xx	4.6.3

NOTE:

1. 'RO' means 'Read Only'.

Base address of SCU and register map are introduced in the followings:

Table 19. Base Address of SCU

Name	Base address
SCU	0x4000_0000

Table 20. SCU Register Map

Name	Offset	Type	Description	Reset value	Reference
SCU_SMR	0x0004	RW	System Mode Register	0x0000_0000	4.6.4
SCU_SCR	0x0008	RW	System Control Register	0x0000_0000	4.6.5
SCU_WUER	0x0010	RW	Wake up source enable register	0x0000_0000	4.6.6
SCU_WUSR	0x0014	RO	Wake up source status register	0x0000_0000	4.6.7
SCU_RSER	0x0018	RW	Reset source enable register	0x0000_0069	4.6.8
SCU_RSSR	0x001C	RW	Reset source status register	0x0000_0080	4.6.9
SCU_PRER1	0x0020	RW	Peripheral reset enable register 1	0x871F_273B	4.6.10
SCU_PRER2	0x0024	RW	Peripheral reset enable register 2	0x88D1_0F3C	4.6.11
SCU_PER1	0x0028	RW	Peripheral enable register 1	0x0000_000F	4.6.12
SCU_PER2	0x002C	RW	Peripheral enable register 2	0x2000_0000	4.6.13
SCU_PCER1	0x0030	RW	Peripheral clock enable register 1	0x0000_000F	4.6.14
SCU_PCER2	0x0034	RW	Peripheral clock enable register 2	0x2000_0000	4.6.15
SCU_PPCLKSR	0x0038	RW	Peripheral clock selection register	0x0000_0000	4.6.16
SCU_CSCR	0x0040	RW	Clock Source Control register	0x0000_0800	4.6.17
SCU_SCCR	0x0044	RW	System Clock Control register	0x0000_0000	4.6.18
SCU_CMR	0x0048	RW	Clock Monitoring register	0x0000_0090	4.6.19
SCU_NMIR	0x004C	RW	NMI control register	0x0000_0000	4.6.20
SCU_COR	0x0050	RW	Clock Output Control register	0x0000_000F	4.6.21
SCU_PLLCON	0x0060	RW	PLL Control register	0x0000_0000	4.6.22

Table 20. SCU Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
SCU_VDCCON	0x0064	RW	VDC Control register	0x0402_887F	4.6.23
SCU_PDRCON	0x0068	RW	PDR Control register	0x0000_0000	4.6.24
SCU_LSICON	0x006C	RW	Internal Ring OSC Control Register	0x0000_0001	4.6.25
SCU_EOSCR	0x0080	RW	External Oscillator control register	0x0000_1014	4.6.26
SCU_EMODR	0x0084	RW	External mode pin read register	0x0000_0000	4.6.27
SCU_RSTDBCR	0x0088	RW	Pin Reset Debounce Control Register	0x0000_0000	4.6.28
SCU_MCCR1	0x0090	RW	Misc. Clock Control register 1	0x0000_0000	4.6.29
SCU_MCCR2	0x0094	RW	Misc. Clock Control register 2	0x0000_0000	4.6.30
SCU_MCCR3	0x0098	RW	Misc. Clock Control register 3	0x0000_0000	4.6.31
SCU_MCCR4	0x009C	RW	Misc. Clock Control register 4	0x0000_0000	4.6.32
SCU_MCCR5	0x00A0	RW	Misc. Clock Control register 5	0x0000_0000	4.6.33
SCU_MCCR6	0x00A4	RW	Misc. Clock Control register 6	0x0000_0000	4.6.34

NOTES:

1. 'RO' means 'Read Only'.
2. 'RW' means 'Read and Write'.

Base address of LVI/LVR unit and register map are introduced in the followings:

Table 21. Base Address of LVI/LVR

NAME	BASE ADDRESS
SCULV(LVI/LVR)	0x4000_5100

Table 22. LVI/LVR Register Map

Name	Offset	Type	Description	Reset value	Reference
SCULV_LVICR	0x0000	RW	Low Voltage Indicator Control Register	0x0000_0000	4.6.35
SCULV_LVRCR	0x0004	RW	Low Voltage Reset Control Register	0x0000_0000	4.6.36
SCULV_LVRCNFIG	0x0008	RW	Configuration for Low Voltage Reset	0x0000_000F	4.6.37

NOTES:

1. 'RO' means 'Read Only' and 'RW' means 'Read and Write'.
2. 'RC' means 'Read and Write 1 Clear'.

4.6.1 SCUCC_VENDORID: vendor ID register

SCUCC_VENDORID register shows the vendor identification information. This is a 32-bit read-only register.

SCUCC_VENDORID=0x4000_F000

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
VENDID			
0x4142_4F56			
RO			

31 0	VENDID	Vendor Identification bits. 0x4142_4F56

4.6.2 SCUCC_CHIPID: chip ID register

SCUCC_CHIPID register shows chip identification information. This is a 32-bit read-only register.

SCUCC_CHIPID=0x4000_F004

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
CHIPID			
0x4D31A006 or 0x4D31A007			
RO			

31 0	CHIPID	Chip Identification bits.	
		0x4D31A006	128k bytes flash memory for program
		0x4D31A007	64k bytes flash memory for program

4.6.3 SCUCC_REVNR: revision number register

Revision Number register is a 32-bit read-only register. This Register is available at 32/16/8-bit access.

SCUCC_REVNR=0x4000_F008

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			REVNO
0x000000			xx
-			RO

7 0	REVNO	Chip Revision Number. These bits are fixed by manufacturer.

4.6.4 SCU_SMR: system mode register

Current operating mode is shown in this SCU mode register. The previous operating mode will be saved in this register after reset event. There is a VDC On/Off control bit in power down mode.

SCU_SMR=0x4000_0004

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			BGRAON	VDCAON	Reserved	PREVMODE	Reserved
-			0	0	-	00	-
-			RW	RW	-	RO	-

9	BGRAON	BGR Always on select bit in power down mode	
		0	BGR is automatically off entering power down mode
		1	BGR isn't automatically off entering power down mode
8	VDCAON	VDC Always on select bit in power down mode	
		0	VDC is automatically off entering power down mode
		1	VDC isn't automatically off entering power down mode
6 4	PREVMODE	Previous operating mode before current reset event	
		000	Previous operating mode was RUN mode
		001	Previous operating mode was SLEEP mode
		011	Previous operating mode was INIT mode
		110	Previous operating mode was Back-up Power mode
		111	Previous operating mode was Standby or Stop mode

NOTE:

1. Power down modes: STOP, STANDBY, and BACKUP POWER

4.6.5 SCU_SCR: system control register

It is possible to reset MCU as SWRST bit set. System Mode Register is a 32-bit register.

SCU_SCR=0x4000_0008

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
WTIDKY		Reserved		SWRST
0x0000		-		0
WO		-		RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0x9EB3 to these bits, otherwise the write is ignored.	
0	SWRST	Internal soft reset activation bit (check RSER[4] for reset)	
		0	Normal operation
		1	Internal soft reset generated and auto cleared

4.6.6 SCU_WUER: wakeup source enable register

Enable wakeup source when the chip is in the Power Down mode. Wakeup sources which will be used the source of chip wakeup should be enabled in each bit field. If the source is used as a wakeup source, the corresponding bit should be written with '1'. If the source is not used as a wakeup source, the bit should be written with '0'.

SCU_WUER=-0x4000_0010

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		I2C1WUE	I2C0WUE	GPIOFWUE	USART13WUE	USART12WUE	GPIOCWUE	GPIOBWUE	GPIOAWUE	Reserved	COMPWUE	RTCWUE	USART11WUE	USART10WUE	WTWUE	WDTWUE	LVRWUE
-		0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

15	I2C1WUE	Enable wakeup source of I2C1 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
14	I2C0WUE	Enable wakeup source of I2C0 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
13	GPIOFWUE	Enable wakeup source of GPIOF port pin change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
12	USART13WUE	Enable wakeup source of USART13 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
11	USART12WUE	Enable wakeup source of USART12 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
10	GPIOCWUE	Enable wakeup source of GPIOC port pin change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
9	GPIOBWUE	Enable wakeup source of GPIOB port pin change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
8	GPIOAWUE	Enable wakeup source of GPIOA port pin change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
6	COMPWUE	Enable wakeup source of COMP change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
5	RTCWUE	Enable wakeup source of RTC change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
4	USART11WUE	Enable wakeup source of USART11 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
3	USART10WUE	Enable wakeup source of USART10 change event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
2	WTWUE	Enable wakeup source of watch timer event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
1	WDTWUE	Enable wakeup source of watchdog timer event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation

NOTE :

1. RTCWUE, WDTWUE, LVRWUE are no reset in STANDBY mode and BACKUP POWER mode.

4.6.7 SCU_WUSR: wakeup source status register

When the system is waked up by any wakeup source, the wakeup source is identified by reading SCU_WUSR register. When the bit is set to 1, the corresponding wakeup source issues the wake-up signal to SCU. The bit will be cleared when the event source is cleared by the software.

SCU_WUSR=0x4000_0014

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		I2C1WU	I2C0WU	GPIOFWU	USART13WU	USART12WU	GPIOCWU	GPIOBWU	GPIOAWU	PC13WU	COMPWU	RTCWU	USART11WU	USART10WU	WTWU	WDTWU	LVRWU
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

15	I2C1WU	Status of wakeup source of I2C1 change event	
		0	No wakeup event
		1	Wakeup event was generated
14	I2C0WU	Status of wakeup source of I2C0 change event	
		0	No wakeup event
		1	Wakeup event was generated
13	GPIOFWU	Status of wakeup source of GPIOF port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
12	USART13WU	Status of wakeup source of USART13 change event	
		0	No wakeup event
		1	Wakeup event was generated
11	USART12WU	Status of wakeup source of USART12 change event	
		0	No wakeup event
		1	Wakeup event was generated
10	GPIOCWU	Status of wakeup source of GPIOC port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
9	GPIOBWU	Status of wakeup source of GPIOB port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
8	GPIOAWU	Status of wakeup source of GPIOA port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
7	PC13WU	Status of wakeup source of PC13 pin change event (Only Standby Mode)	
		0	No wakeup event
		1	Wakeup event was generated

6	COMPWU	Status of wakeup source of COMP change event	
		0	No wakeup event
		1	Wakeup event was generated
5	RTCWU	Status of wakeup source of RTC change event	
		0	No wakeup event
		1	Wakeup event was generated
4	USART11WU	Status of wakeup source of USART11 change event	
		0	No wakeup event
		1	Wakeup event was generated
3	USART10WU	Status of wakeup source of USART10 change event	
		0	No wakeup event
		1	Wakeup event was generated
2	WTWU	Status of wakeup source of watch timer event	
		0	No wakeup event
		1	Wakeup event was generated
1	WDTWU	Status of wakeup source of watchdog timer event	
		0	No wakeup event
		1	Wakeup event was generated
0	LVRWU	Status of wakeup source of LVR event	
		0	No wakeup event
		1	Wakeup event was generated

4.6.8 SCU_RSER: reset source enable register

A reset source to CPU can be selected by SCU_RSER register. When writing '1' in the bit field of each reset source, the reset source event will be transferred to reset generator. When writing '0' in the bit field of each reset source, the reset source event will be masked and not generate the reset event.

SCU_RSER=0x4000_0018

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				PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVRRST
-				1	1	0	1	0	0	1
-				RW	RW	RW	RW	RW	RW	RW

6	PINRST	External pin reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
5	CPURST	CPU request reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
4	SWRST	Software reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
3	WDTRST	Watchdog Timer reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
2	MCKFRST	MCLK Clock fail reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
1	MOFRST	HSE Clock fail reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
0	LVRRST	LVR reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled

NOTE:

1. PINRST, WDTRST, LVRRST are reset only by LVR reset and VPWRSW POR.

4.6.9 SCU_RSSR: reset source status register

SCU_RSSR register shows reset source information when reset event is occurred. '1' means reset event was exist and '0' means reset event is not exist for the corresponding reset source.

SCU_RSSR=0x4000_001C

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			PORST	PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVDRST
-			1	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7	PORST	Power on reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
6	PINRST	External pin reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
5	CPURST	CPU request reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
4	SWRST	Software reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
3	WDTRST	Watchdog Timer reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
2	MCKFRST	MCLK Clock fail reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status

1	MOFRST	HSE Clock fail reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
0	LVRRST	LVR reset status bit	
		0	Read : Reset from this event was not exist Write : no effect
		1	Read : Reset from this event was occurred Write : Clear the status
NOTE: When reset source is founded, write '1' into the corresponding bit to clear the reset status.			

4.6.10 SCU_PRER1: peripheral reset enable register 1

The reset of each peripheral by event reset, can be masked by user setting. SCU_PRER1/SCU_PRER2 register will control the enable of the event reset. If the corresponding bit is '1', the peripheral corresponded with this bit, accepts the reset event. Otherwise, the peripheral is protected from reset event and maintain current operation.

SCU_PRER1=0x4000_0020

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
WT	Reserved	TIMER21	TIMER20	TIMER30	Reserved	TIMER40	TIMER13	TIMER12	TIMER11	TIMER10	Reserved	GPIOF	Reserved	GPIOC	GPIOB	GPIOA	Reserved	EBI	DMA	Reserved
1	-	1	1	1	-	1	1	1	1	1	-	1	-	1	1	1	-	1	1	-
RW	-	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW	-	RW	RW	RW	-	RW	RW	-

31	WT	WT reset mask
26	TIMER21	TIMER21 reset mask
25	TIMER20	TIMER20 reset mask
24	TIMER30	TIMER30 reset mask
20	TIMER40	TIMER40 reset mask
19	TIMER13	TIMER13 reset mask
18	TIMER12	TIMER12 reset mask
17	TIMER11	TIMER11 reset mask
16	TIMER10	TIMER10 reset mask
13	GPIOF	GPIOF reset mask
10	GPIOC	GPIOC reset mask
9	GPIOB	GPIOB reset mask
8	GPIOA	GPIOA reset mask
5	EBI	EBI reset mask
4	DMA	DMA reset mask

4.6.11 SCU_PRER2: peripheral reset enable register 2

SCU_PRER2 is a 32-bit register.

SCU_PRER2=0x4000_0024

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
CRC	Reserved	RTC	Reserved	TSENSE	Reserved			COMP	DAC	Reserved	ADC	Reserved			USB	Reserved				USART13	USART12	USART11	USART10	Reserved		I2C1	I2C0	SPI21	SPI20	Reserved	
1	-	0	-	1	-			1	1	-	1	-			1	-				1	1	1	1	-		1	1	1	1	-	
RW	-	RW	-	RW	-			RW	RW	-	RW	-			RW	-				RW	RW	RW	RW	-		RW	RW	RW	RW	-	

31	CRC	CRC reset mask
29	RTC	RTC reset mask
27	TSENSE	TSENSE reset mask
23	COMP	COMP reset mask
22	DAC	DAC reset mask
20	ADC	ADC reset mask
16	USB	USB reset mask
11	USART13	USART13 reset mask
10	USART12	USART12 reset mask
9	USART11	USART11 reset mask
8	USART10	USART10 reset mask
5	I2C1	I2C1 reset mask
4	I2C0	I2C0 reset mask
3	SPI21	SPI21 reset mask
2	SPI20	SPI20 reset mask

4.6.12 SCU_PER1: peripheral enable register1

Before using a peripheral unit, it must be activated by writing '1' to a corresponding bit in SCU_PER1/ SCU_PER2 register. Before the activation, the peripheral will stay in reset state.

All peripherals are enabled by default. To disable the peripheral unit, write '0' to the corresponding bit in the SCU_PER1/ SCU_PER2 register, and then the peripheral goes under reset state.

SCU_PER1=0x4000_0028

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
WT	Reserved				TIMER21	TIMER20	TIMER30	Reserved			TIMER40	TIMER13	TIMER12	TIMER11	TIMER10	Reserved		GPIOF	Reserved		GPIOC	GPIOB	GPIOA	Reserved		EBI	DMA	Reserved			
0	-				0	0	0	0			0	0	0	0	0	-		0	-		0	0	0	-		0	0	-			
RW	-				RW	RW	RW	RW			RW	RW	RW	RW	RW	-		RW	-		RW	RW	RW	-		RW	RW	-			

31	WT	WT function enable
26	TIMER21	TIMER21 function enable
25	TIMER20	TIMER20 function enable
24	TIMER30	TIMER30 function enable
20	TIMER40	TIMER40 function enable
19	TIMER13	TIMER13 function enable
18	TIMER12	TIMER12 function enable
17	TIMER11	TIMER11 function enable
16	TIMER10	TIMER10 function enable
13	GPIOF	GPIOF function enable
10	GPIOC	GPIOC function enable
9	GPIOB	GPIOB function enable
8	GPIOA	GPIOA function enable
5	EBI	EBI function enable
4	DMA	DMA function enable

4.6.13 SCU_PER2: peripheral enable register 2

Peripheral enable register 2 is a 32-bit register.

SCU_PER2=0x4000_002C

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
CRC	Reserved	RTC	Reserved	TSENSE	Reserved			COMP	DAC	Reserved	ADC	Reserved			USB	Reserved				USART13	USART12	USART11	USART10	Reserved		I2C1	I2C0	SPI21	SPI20	Reserved	
0	-	1	-	0	-			0	0	-	0	-			0	-				0	0	0	0	-		0	0	0	0	-	
RW	-	RW	-	RW	-			RW	RW	-	RW	-			RW	-				RW	RW	RW	RW	-		RW	RW	RW	RW	-	

31	CRC	CRC function enable
29	RTC	RTC function enable
27	TSENSE	TSENSE function enable
23	COMP	COMP function enable
22	DAC	DAC function enable
20	ADC	ADC function enable
16	USB	USB function enable
11	USART13	USART13 function enable
10	USART12	USART12 function enable
9	USART11	USART11 function enable
8	USART10	USART10 function enable
5	I2C1	I2C1 function enable
4	I2C0	I2C0 function enable
3	SPI21	SPI21 function enable
2	SPI20	SPI20 function enable

4.6.14 SCU_PCER1: peripheral clock enable register 1

To use peripheral unit, its clock should be activated by writing '1' to the corresponding bit in the SCU_PCER1 register. Without enabling the clock, the peripheral won't operate properly.

To stop the clock of the peripheral, write '0' to the corresponding bit in the SCU_PCER1/ SCU_PCER2 register, then the clock will be stopped.

SCU_PCER1=0x4000_0030

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
WT	Reserved				TIMER21	TIMER20	TIMER30	Reserved			TIMER40	TIMER13	TIMER12	TIMER11	TIMER10	Reserved		GPIOF	Reserved		GPIOC	GPIOB	GPIOA	Reserved		EBI	DMA	Reserved			
0	-				0	0	0	-			0	0	0	0	0	-		0	-		0	0	0	-		0	0	-			
RW	-				RW	RW	RW	-			RW	RW	RW	RW	RW	-		RW	-		RW	RW	RW	-		RW	RW	-			

31	WT	WT clock enable
26	TIMER21	TIMER21 clock enable
25	TIMER20	TIMER20 clock enable
24	TIMER30	TIMER30 clock enable
20	TIMER40	TIMER40 clock enable
19	TIMER13	TIMER13 clock enable
18	TIMER12	TIMER12 clock enable
17	TIMER11	TIMER11 clock enable
16	TIMER10	TIMER10 clock enable
13	GPIOF	GPIOF clock enable
10	GPIOC	GPIOC clock enable
9	GPIOB	GPIOB clock enable
8	GPIOA	GPIOA clock enable
5	EBI	EBI clock enable
4	DMA	DMA clock enable

4.6.15 SCU_PCER2: peripheral clock register 2

To use peripheral unit, its clock should be activated by writing '1' to the corresponding bit in the SCU_PCER2 register.

SCU_PCER2=0x4000_0034

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
CRC	Reserved	RTC	Reserved	TSENSE	Reserved			COMP	DAC	Reserved	ADC	Reserved			USB	Reserved				USART13	USART12	USART11	USART10	Reserved		I2C1	I2C0	SPI21	SPI20	Reserved	
0	-	1	-	0	-			0	0	-	0	-			0	-				0	0	0	0	-		0	0	0	0	-	
RW	-	RW	-	RW	-			RW	RW	-	RW	-			RW	-				RW	RW	RW	RW	-		RW	RW	RW	RW	-	

31	CRC	CRC clock enable
29	RTC	RTC clock enable
27	TSENSE	TSENSE clock enable
23	COMP	COMP clock enable
22	DAC	DAC clock enable
20	ADC	ADC clock enable
16	USB	USB clock enable
11	USART13	USART13 clock enable
10	USART12	USART12 clock enable
9	USART11	USART11 clock enable
8	USART10	USART10 clock enable
5	I2C1	I2C1 clock enable
4	I2C0	I2C0 clock enable
3	SPI21	SPI21 clock enable
2	SPI20	SPI20 clock enable

4.6.16 SCU_PPCLKSR: peripheral clock selection register

SCU_PPCLKSR register is a 32-bit register. This register is available at 32/16/8-bit access.

SCU_PPCLKSR=0x4000_0038

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		T1xCLK	Reserved	T20CLK	T21CLK	Reserved	T30CLK	T40CLK	Reserved	USBCLK	Reserved	RTCCLK	WTCLK	Reserved	WDTCLK
-		0	-	0	0	-	0	0	-	0	-	0	00	-	0
-		RW	-	RW	RW		RW	RW	-	RW	-	RW	RW	-	RW

22	T1xCLK	Timer 1x Clock Selection bit
		0 SCU_MCCR1 Timer1x clock
		1 PCLK clock
20	T20CLK	Timer 20 Clock Selection bit
		0 SCU_MCCR2 Timer20 clock
		1 PCLK clock
19	T21CLK	Timer 21 Clock Selection bit
		0 SCU_MCCR6 Timer21 clock
		1 PCLK clock
17	T30CLK	Timer 30 Clock Selection bit
		0 SCU_MCCR2 Timer30 clock
		1 PCLK clock
16	T40CLK	Timer 40 Clock Selection bit
		0 SCU_MCCR6 Timer40 bit
		1 PCLK clock
9	USBCLK	USB Clock Selection bit
		0 SCU_MCCR5 USB clock
		1 PCLK clock
5	RTCCLK	RTC Clock Selection bit
		0 LSI40K clock
		1 SCU_MCCR5 RTC clock
4 3	WTCLK	Watch Timer Clock Selection bit
		00 SCU_MCCR3 WT clock
		01 LSE clock
		10 WDTRC clock
		11 LSI40K clock
		NOTE: These bits should be changed during the WTEN bit of watch timer control register (WTCR) is "0b".
0	WDTCLK	Watch-dog Timer Clock Selection bit
		0 LSI40K clock
		1 SCU_MCCR3 WDT clock

NOTE:

1. RTCCLK and WDTCLK can be reset only by LVR reset and VPWRSW POR.

4.6.17 SCU_CSCR: clock source control register

A31G32x series have multiple clock sources to generate internal operating clocks. Each clock source can be controlled by SCU_CSCR register.

SCU_CSCR=0x4000_0040

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
WTIDKY		LSECON	LSICON	HSICON	HSECON
0x0000		0000	1000	0000	0000
WO		RW	RW	RW	RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0xA507 to these bits, otherwise the write is ignored.	
15 12	LSECON	External crystal sub oscillator control	
		0XXX	Disable external sub crystal oscillator
		1000	Enable external sub crystal oscillator
		1001	Enable external sub crystal oscillator divide by 2
		1010	Enable external sub crystal oscillator divide by 4
		1100	Enable external sub crystal oscillator in Power Down mode
		1101	Enable external sub crystal oscillator divide by 2 in Power Down mode
		1110	Enable external sub crystal oscillator divide by 4 in Power Down mode
		Other	Reserved
11 8	LSICON	Low speed internal oscillator control	
		0XXX	Disable low speed internal oscillator
		1000	Enable low speed internal oscillator
		1001	Enable low speed internal oscillator divide by 2
		1010	Enable low speed internal oscillator divide by 4
		Other	Reserved
7 4	HSICON	High speed internal oscillator control	
		0XXX	Disable high speed internal oscillator
		1000	Enable high speed internal oscillator
		1001	Enable high speed internal oscillator divide by 2
		1010	Enable high speed internal oscillator divide by 4
		1011	Enable high speed internal oscillator divide by 8
		1100	Enable high speed internal oscillator divide by 16
		1101	Enable high speed internal oscillator divide by 32
		1111	Reserved
3 0	HSECON	External crystal main oscillator control	
		0XXX	Disable external main crystal oscillator
		1000	Enable external main crystal oscillator
		1001	Enable external main crystal oscillator divide by 2
		1010	Enable external main crystal oscillator divide by 4
		Other	Reserved

NOTES:

1. SCU_CSCR[15:12] are reset only by LVR reset and VPWRSW POR.
2. To use LSE in Power down Mode, Select 1100b' in SCU_CSCR[15:12].

4.6.18 SCU_SCCR: system clock control register

Selected system clock source in SCU_SCCR becomes MCLK. Before changing clock, clock sources have to be alive by SCU_CSCR register.

SCU_SCCR=0x4000_0044

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
WTIDKY		Reserved		PLLINCLKSE	MCLKSEL
0x0000		-		0	00
WO		-		RW	RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0x570A to these bits, otherwise the write is ignored.	
2	PLLINCLKSEL	PLL input source select register	
		0	HSI clock is used as PLLINCLK clock
		1	HSE clock is used as PLLINCLK clock
1 0	MCLKSEL	System clock select register	
		00	Internal ring oscillator(750kHz)
		01	LSE XTAL (32kHz)
		10	PLL bypassed clock
		11	PLL output clock

NOTE:

1. When change MCLKSEL, both clock sources should be alive. Ex) Both of HSI and HSE should be alive, otherwise the chip will malfunction.

4.6.19 SCU_CMR: clock monitoring register

Clock can be monitored by LSI for security purpose.

SCU_CMR=0x4000_0048

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		MCLKREC	Reserved	LSEMNT	LSEIE	LSEFAIL	LSESTS	MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	HSEMNT	HSEIE	HSEFAIL	HSESTS
-		0	-	0	0	0	0	1	0	0	1	0	0	0	0
-		RW	-	RW	RW	RC	RO	RW	RW	RC	RO	RW	RW	RC	RO

15	MCLKREC	MCLK fail auto recovery	
		0	MCLK is changed to LSI by default when MCLKFAIL issued
		1	MCLK auto recovery is disabled
11	LSEMNT	External sub oscillator monitoring enable	
		0	External sub oscillator monitoring disabled
		1	External sub oscillator monitoring enabled
10	LSEIE	External sub oscillator fail interrupt enable	
		0	External sub oscillator fail interrupt disabled
		1	External sub oscillator fail interrupt enabled
9	LSEFAIL	External sub oscillator fail interrupt	
		0	External sub oscillator fail interrupt not occurred
		1	Read : External sub oscillator fail interrupt is pending Write : Clear pending interrupt
8	LSESTS	External sub oscillator status	
		0	Not oscillate
		1	External sub oscillator is working normally
7	MCLKMNT	MCLK monitoring enable	
		0	MCLK monitoring disabled
		1	MCLK monitoring enabled
6	MCLKIE	MCLK fail interrupt enable	
		0	MCLK fail interrupt disabled
		1	MCLK fail interrupt enabled
5	MCLKFAIL	MCLK fail interrupt	
		0	MCLK fail interrupt not occurred
		1	Read : MCLK fail interrupt is pending Write : Clear pending interrupt
4	MCLKSTS	MCLK clock status	
		0	No clock is present on MCLK
		1	Clock is present on MCLK
3	HSEMNT	External main oscillator monitoring enable	
		0	External main oscillator monitoring disabled
		1	External main oscillator monitoring enabled
2	HSEIE	External main oscillator fail interrupt enable	
		0	External main oscillator fail interrupt disabled
		1	External main oscillator fail interrupt enabled
1	HSEFAIL	External main oscillator fail interrupt	
		0	External main oscillator fail interrupt not occurred
		1	Read : External main oscillator fail interrupt is pending Write : Clear pending interrupt
0	HSESTS	External main oscillator status	
		0	Not oscillate
		1	External main oscillator is working normally

4.6.20 SCU_NMIR: NMI control register

SCU_NMIR is a non-maskable interrupt configuration register which can be set by software. There are three types of interrupt sources from WDT and SCU.

It will jump to NMI handler if a selected NMI event occurred and it must check event status. For clearing occurred status, it should clear the interrupt flags of that peripheral occurred.

Write access key is required 0xA32C on SCU_NMIR[31:16] when write register.

SCU_NMIR=0x4000_004C

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
ACCESSCODE		Reserved	WDTINTSTS	MCLKFAILSTS	LVRSTS	Reserved	WDTINTEN	MCLKFAILEN	LVREN
-		-	0	0	0	-	0	0	0
WO		-	RO	RO	RO	-	RW	RW	RW

31 16	ACCESSCODE	This field enables writing access to this register. Writing 0xA32C is to enable writing.
10	WDTINTSTS	WDT Interrupt condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
9	MCLKFAILSTS	MCLK Fail condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
8	LVRSTS	LVR condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
2	WDTINTEN	WDT Interrupt condition enable for NMI interrupt
		0 Disable
		1 Enable
1	MCLKFAILEN	MCLK Fail condition enable for NMI interrupt
		0 Disable
		1 Enable
0	LVREN	LVR Fail condition enable for NMI interrupt
		0 Disable
		1 Enable

4.6.21 SCU_COR: clock output register

A31G32x series can drive a clock from internal MCLK clock with a dedicated post divider. To use CLKO output function, it should be set as CLKO that has output mode in PF0 Pin Mux. SCU_COR register is an 8-bit register.

SCU_COR=0x4000_0050

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				CLKOEN	CLKODIV
-				0	1111
-				RW	RW

4	CLKOEN	Clock output enable	
		0	CLKO is disabled and stay "L" output
		1	CLKO Is enabled
3 0	CLKODIV	Clock output divider value CLKO = MCLK (CLKODIV = 0) $\mathrm{CLKO} = \frac{MCLK}{2 * (\mathrm{CLKODIV} + 1)} \quad (CLKODIV > 0)$	

4.6.22 SCU_PLLCON: PLL control register

Integrated PLL will synthesize high speed clock for extremely high performance of CPU. The PLL is controlled by setting the register. PLL Control Register is 32-bit register.

SCU_PLLCON=0x4000_0060

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
PLLLOCK	Reserved	PLLRSTB	PLLEN	BYPASSB	Reserved	PREDIV	POSTRIV1	POSTDIV2	OUTDIV
0	-	0	0	0	-	000	00000000	0000	0000
RO	-	RW	RW	RW	-	RW	RW	RW	RW

31	PLLLOCK	PLLLOCK status	
		0	PLL is not locked
		1	PLL is locked
23	PLLRSTB	PLL reset	
		0	PLL reset is asserted
		1	PLL reset is negated
22	PLLEN	PLL enable	
		0	PLL is disabled
		1	PLL is enabled
21	BYPASSB	PLLINCLK bypass	
		0	FOUT is bypassed as PLLINCLK
		1	FOUT is PLL output
18 16	PREDIV	PLLINCLK predivider (R)	
		0~7	PLLINCLK divided by (PREDIV + 1), (PLLINCLK/1 to PLLINCLK/8)
15 8	POSTDIV1	Feedback control 1 (N1)	
		0x00	N1 = 0 (N1 + 1)
		0xFF	N1 = 255 (N1 + 1)
7 4	POSTDIV2	Feedback control 2 (N2)	
		0x0	N2 = 0 (N2 + 1)
		0xF	N2 = 15 (N2 + 1)
3 0	OUTDIV	Output divider control (P)	
		0x0	P = 0 (P+1)
		0xF	P = 15 (P+1)

NOTES:
Bit 20 of PLLCON must be kept at 0.
When PLLEN is set to '1', PLLRSTB is set to '1' and PLL Divider count after at least 1us.
Wait more than 190us for PLL Lock Time, then check the PLL Lock flag(Bit 31).
At power down(Stop/Standby) mode, set PLLRSTB to '0', PLLEN to '0'
Bits of OUTDIV is not recommended to set to '1'.
Output calculation formula is as followings:

$$F_{OUT} = \frac{PLLINCLK \times (N_1 + 1)}{(R + 1) \times (N_2 + 1) \times (P + 1)}$$

Symbol	Description
R	Pre Divider Counter Value
N_1	Post Divider1 Counter Value
N_2	Post Divider2 Counter Value
P	Output Divider Counter Value

Calculating PLL output frequency value

PLL of A31G32x series can accurately set the output frequency, F_{OUT}, in 1MHz increments. The formula for the F_{IN} input to the F_{VCO} input of the PLL is as follows, and the input range of the F_{IN} frequency is between 1MHz and 3MHz. However, it is recommended to set the input frequency (FIN) to 2MHz as much as possible.

$$\mathrm{FIN} = \frac{\mathrm{PLLINCLK}}{(\mathrm{R}+1)}, \qquad \text{Where } 1\mathrm{MHz} \le \mathrm{FIN} \le 3\mathrm{MHz}\ (\text{Recommanded FIN} = 2\mathrm{MHz})$$

At this time, the range of F_{VCO} output frequency should be set to 288MHz or less, and the calculation formula is as follows.

$$\mathrm{F_{VCO}} = \mathrm{FIN} \times (\mathrm{N_1} + 1), \quad \mathrm{VCO} \le 288\mathrm{MHz}$$

As a result, the final frequency of PLL, F_{OUT}, can be obtained from the formula below using the formula above.

$$\mathrm{F_{OUT}} = \frac{\mathrm{PLLINCLK} \times (\mathrm{N_1}+1)}{(\mathrm{R}+1) \times (\mathrm{N_2}+1) \times (\mathrm{P}+1)} = \frac{\mathrm{FIN} \times (\mathrm{N_1}+1)}{(\mathrm{N_2}+1) \times (\mathrm{P}+1)}$$

4.6.23 SCU_VDCCON: VDC control register

On chip VDC control register. VDCTRIM is used for the trim value of VDC output. VDCWDLY value can be written with writing '1' to VDCWDLY_WEN bit simultaneously.

SCU_VDCCON=0x4000_0064

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
VDC33 WTIDKY	VDC33_PDBGR	VDC33_STOP	VDC33_BYPSBS	VDC33_LOCK	VDC15 WTIDKY	Reserved	VDC15_STBY	Reserved	VDC33_PD	VDCWDLY_WEN	VDCWDLY
0000	0	1	0	0	0000	-	0	-	0	0	0x7F
WO	RW	RW	RW	RW	WO	-	RW	-	RW	WO	RW

31 28	VDC33 WTIDKY	VDC33 Write Identification Key On writes, write 0xA to these bits, otherwise the write is ignored.	
27	VDC33_PDBGR	VDC33 BGR Power Down Signal	
		0	BGR Alive
		1	BGR Power Down
26	VDC33_STOP	VDC33 STOP Mode Enable Signal	
		0	RUN mode
		1	STOP mode
25	VDC33_BYPASS	VDC33 Bypass Mode Enable Signal	
		0	5V Bypass Enable
		1	Disable
24	VDC33_LOCK	VDC Control signal for *BGR stabilization	
		0	Select BMR when BGR off
		1	Select BGR after stabilization time when BGR is on
23 20	VDC15 WTIDKY	VDC15 Write Identification Key On writes, write 0x5 to these bits, otherwise the write is ignored.	
18	VDC15_STBY	Power down mode selection signal when CPU enters deepsleep mode	
		0	Stop Mode
		1	Standby Mode
9	VDC33_PD	VDC33 Power Down Signal	
		0	VDC33 is not Power Down Mode
		1	VDC33 is Power Down Mode
8	VDCWDLY_WEN	VDCWDLY value write enable. VDCWDLY value can be written with writing '1' to VDCWDLY_WEN bit simultaneously.	
7 0	VDCWDLY	VDC warm-up delay count value. When SCU is waked up from power down mode, the warm-up delay is inserted for VDC output being stabilized. The amount of delay can be defined with this register value 7F : 4msec	
NOTE: Reserved bits should never be modified.			

Table 23. VDC33 Mode Description

—	VDC33_PDBGR	VDC33_STOP	VDC33_BYPSB	VDC33_LOCK	VDC33_PD	Output
RUN	0	0	1	1	0	3.3V
STOP	1	1	1	0	0	3.3V
BYPASS	1	X	0	0	0	VDD
PD	1	0	1	x	1	Floating

NOTES:

1. In power down mode, VDC33 must be configured in STOP or BYPASS mode to reduce current consumption.
2. When supplying external power 3.3V to USB, VDC33 must be configured in PD mode.
3. When VDC33 BGR is used as a single ended receiver hysteresis reference voltage in VDC33 PD mode, PDBGR should be set to 1.

4.6.24 SCU_PDRCON: PDR control register

SCU_VDCCON=0x4000_0064

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		PDRCON WTIDKY	PDREN	Reserved
-		0x0	0	-
-		WO	RW	-

11 8	PDRCON WTIDKY	PDRCON Write Identification Key On writes, write 0xA to these bits, otherwise the write is ignored.	
7	PDREN	PDR Enable	
		0	PDR Disable
		1	PDR Enable

NOTE:

1. When using the backup power mode, disable LVR (SCU_LVRCR, SCU_LVRCNFIG) and LVR reset (SCU_RSER).

4.6.25 SCU_LSICON: Low speed internal OSC control register

SCU_LSICON=0x4000_006C

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				EN_LDO	SKIP
-				0	1
-				RW	RW

1	EN_LDO	Internal LDO On/Off	
		0	Disable
		1	Enable
0	SKIP_LS	Internal Level Shifter control signal	
		0	Enable
		1	Disable

4.6.26 SCU_EOSCR: External oscillator control register

External main crystal oscillator has two characteristics. For the noise immunity, NMOS amp type is recommended and for the low power characteristic, INV amp type is recommended. SCU_EOSCR register is a 16-bit register.

SCU_EOSCR=0x4000_0080

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		ESEN	Reserved	ESISEL	Reserved	ESSMT	EMEN	Reserved	ISEL	NCOPT	Reserved	NCSKIP
-		0	-	01	-	0	0	-	01	01	-	0
-		W	-	RW	-	RW	W	-	RW	RW	-	RW

15	ESEN	Write enable for External LSE	
		0	Write access disabled
		1	Write access enabled
13 12	ESISEL	Select current for External LSE	
		00	High
		01	Mid-High (default)
		10	Mid-Low
		11	Low
8	ESSMT	Select Schmitt trigger	
		0	Schmitt trigger bypass
		1	Schmitt trigger turn on

7	EMEN	Write enable for External HSE	
		0	Write access disabled
		1	Write access enabled
5 4	ISE	Select current for External HSE	
		00	High
		01	Mid-High (default)
		10	Mid-Low
		11	Low
3 2	NCOPT	Noise Cancel delay Option for External HSE	
		00 01 10 11	25ns (12MHz < HSE < 16MHz) 20ns (8MHz < HSE < 12MHz) 15ns (4MHz < HSE < 8MHz) 10ns (2MHz < HSE < 4MHz)
0	NCSKIP	Noise Cancel SKIP enable for External HSE	
		0	Disable
		1	Enable (Noise Cancel skipped)

4.6.27 SCU_EMODR: External mode status register

SCU_EMODR register shows external mode pin status while booting. This register is an 8-bit register.

SCU_EMODR=0x4000_0084

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				BOOT
-				-
-				R

0	BOOT	BOOT pin level	
		0	BOOT pin is low
		1	BOOT pin is high

4.6.28 SCU_RSTDBCR: pin reset debounce control register

SCU_ RSTDBCR=0x4000_0088

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	
WTIDKY		Reserved	CLKCNT	Reserved	EN
0x0000		-	0x00	-	0
WO		-	RW	-	RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0x0514 to these bits, otherwise the write is ignored.	
13 8	CLKCNT	Noise Cancel delay Option for LSI 750kHz	
		N	N clock checking for debounce by LSI (750kHz)
0	EN	Pin reset debounce enable bit	
		0	Disable
		1	Enable

NOTE: If a user wants to operate pin reset debounce, the user must enable LSI (750kHz). Because pin reset debounce uses LSI for clock source.
The port debounce should be disabled before Power Down mode.

4.6.29 SCU_MCCR1: Miscellaneous clock control register 1

A31G32x series can drive a clock from internal MCLK clock with dedicated post divider. STCSEL bits and STCDIV bits of MCCR1 are used as SYSTICK external clock source. TEXT1CSEL bits and TEXT1DIV bits of SCU_MCCR1 are used as TIMER1n external clock source. Register SCU_MCCR1 is a 32-bit register.

SCU_MCCR1=0x4000_0090

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	TEXT1CSEL	TEXT1DIV	Reserved	STCSEL	SYSTICKDIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	TEXT1CSEL	TIMER1n EXT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
23 16	TEXT1DIV	TIMER1n EXT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing TEXT1CSEL
10 8	STCSEL	SYSTIC Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
7 0	STDIV	SYSTICK Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing STCSEL.

4.6.30 SCU_MCCR2: Miscellaneous clock control register 2

A31G32x series can drive a clock from internal MCLK clock with a dedicated post divider. TEXT2CSEL bits and TEXT2DIV bits of MCCR2 are used as a TIMER20 external clock source. TEXT3CSEL bits and TEXT3DIV bits of SCU_MCCR2 are used as a TIMER30 external clock source. Register SCU_MCCR2 is a 32-bit register.

SCU_MCCR2=0x4000_0094

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	TEXT3CSEL	TEXT3DIV	Reserved	TEXT20CSEL	TEXT20DIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	TEXT3CSEL	TIMER 30 EXT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
23 16	TEXT3DIV	TIMER 30 EXT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing TEXT3DIV
10 8	TEXT20CSEL	TIMER 20 EXT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
7 0	TEXT20DIV	TIMER 20 EXT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing TEXT20CSEL.

4.6.31 SCU_MCCR3: Miscellaneous clock control register 3

A31G32x series can drive a clock from an internal MCLK clock with a dedicated post divider. WDTCSEL bits and WDTDIV bits of a MCCR3 are used as WDT external clock source. WTEXTCSEL bits and WTEXTCDIV bits of SCU_MCCR3 are used as a WT external clock source. This register is a 32-bit register.

SCU_MCCR3=0x4000_0098

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	WTEXTCSEL	WTEXTCDIV	Reserved	LSI40kHz	WDTCSEL	WDTDIV
	0x0	0x00		0	0x0	0x00
	RW	RW		RW	RW	RW

26 24	WTEXTCSEL	WT External Clock source select bit 0xx LSI 750kHz 011 LSE (32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
23 16	WTEXTCDIV	WT External Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing WTEXTCSEL
11	LSI40kHz	LSI40kHz Enable 0 Disable 1 Enable
10 8	WDTCSEL	WDT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
7 0	WDTDIV	WDT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing WDTCSEL.

4.6.32 SCU_MCCR4: Miscellaneous clock control register 4

A31G32x series can drive a clock from an internal MCLK clock with a dedicated post divider. PD0CSEL bits and PD0DIV bits of SCU_MCCR4 are used as PA, PB, PC, PF Debounce Clock source. This register is a 32-bit register.

SCU_MCCR4=0x4000_009C

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			PD0CSEL	PD0DIV
-			0x0	0x00
-			RW	RW

10 8	PD0CSEL	Debounce Clock for PORT source select bit (PA,PB,PC,PF) 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
7 0	PD0DIV	PORT Debounce Clock N divider (PA,PB,PC,PF) 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing PD0CSEL.

4.6.33 SCU_MCCR5: Miscellaneous clock control register 5

A31G32x series can drive a clock from an internal MCLK clock with a dedicated post divider. PD0CSEL bits and PD0DIV bits of MCCR5 are used as PA, PB, PC Debounce Clock source. USBCSEL bits and USBCDIV bits of MCCR5 are used as a USB Clock source. This register is a 32-bit register.

SCU_MCCR5=0x4000_00A0

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	USBCSEL	Reserved	USBEXTCLKEN	TSREFCLK_SEL	TSSENSECLK_SEL	LSITS_EN	RTCCSEL	RTCDIV
-	0x0	-		0x0	0x0		0x0	0x00
-	RW	-	RW	RW	RW	RW	RW	RW

26 24	USBCSEL	USB Clock source select bit	
		0xx	Reserved
		100	MCLK (bus clock)
		101	HSI 48MHz
		110	Reserved
		111	PLL Clock
16	USBEXTCLKEN	USB External clock enable	
		0	Disable
		1	Enable
		To change the value, set 0x0 first without changing USBCSEL	
15 14	TSREFCLK_SEL	Temp Sensor Reference clock select bit Note) When reference clock is sub-osc, reference and sense clock are changed to each other.	
		00	HSI_I (Output clock set by CSCR from HSI)
		01	MCLK
		10	HSE
		11	LSE
13 12	TSSENSECLK_SEL	Temp Sensor Sense clock select bit	
		00	LSITS
		01	LSI750kHz
		10	LSI40kHz
		11	HSI_I (Output clock set by CSCR from HSI)
11	LSITS_EN	LSITS Enable	
		0	Disable
		1	Enable
10 8	RTCCSEL	RTC Clock source select bit	
		0xx	LSI 750kHz
		011	LSE (32kHz)
		100	MCLK (bus clock)
		101	HSI 48MHz
		110	HSE
		111	PLL Clock

7 0	RTCDIV	RTC Clock N divider 8'h0 disabled 8'hN (selected clock) / N

4.6.34 SCU_MCCR6: Miscellaneous clock control register 6

A31G32x series can drive a clock from an internal MCLK clock with a dedicated post divider. TEXT21CSEL bits and TEXT21DIV bits of MCCR6 are used as a TIMER21 external clock source. TEXT4CSEL bits and TEXT4DIV bits of SCU_MCCR6 are used as a TIMER40 external clock source. This register is a 32-bit register.

SCU_MCCR2=0x4000_00A4

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	TEXT21CSEL	TEXT21DIV	Reserved	TEXT40CSEL	TEXT40DIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	TEXT21CSEL	TIMER 21 EXT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
23 16	TEXT21DIV	TIMER 21 EXT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing TEXT2DIV
10 8	TEXT4CSEL	TIMER 40 EXT Clock source select bit 0xx LSI 750kHz 011 LSE(32kHz) 100 MCLK (bus clock) 101 HSI 48MHz 110 HSE 111 PLL Clock
7 0	TEXT4DIV	TIMER 40 EXT Clock N divider 8'h0 disabled 8'hN (selected clock) / N To change the value, set 0x0 first without changing TEXT40CSEL

4.6.35 SCULV_LVICR: Low voltage indicator control register

SCULV_LVICR is a 32-bit register, and available at 32/16/8-bit access.

SCULV_LVICR=0x4000_5100

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			LVIEN	Reserved	LVINTEN	LVIFLAG	LVIVS
-			0	-	0	0	0000
-			RW	-	RW	RW	RW

7	LVIEN	LVI Enable bit.	
		0	Disable low voltage indicator.
		1	Enable low voltage indicator.
5	LVINTEN	LVI Interrupt Enable bit.	
		0	Disable low voltage indicator interrupt.
		1	Enable low voltage indicator interrupt.
4	LVIFLAG	LVI Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3 0	LVIVS	LVI Voltage Selection bits.	
		0000	1.63V
		0001	1.72V
		0010	1.82V
		0011	1.94V
		0100	2.03V
		0101	2.16V
		0110	2.35V
		0111	2.52V
		1000	2.72V
		1001	3.10V
		1010	3.24V
		1011	3.66V
		1100	3.79V
		1101	4.11V
		1110	4.28V
		1111	4.57V

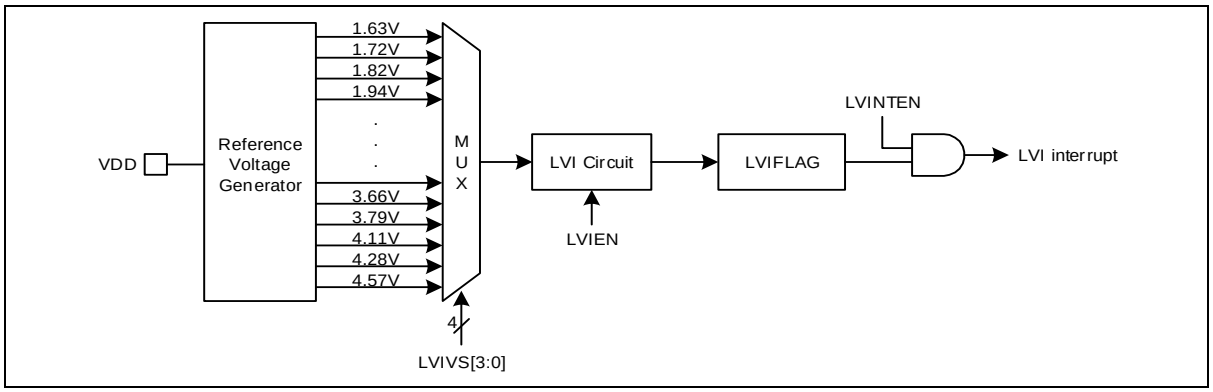


Figure 29. LVI Block Diagram

4.6.36 SCULV_LVRCR: Low voltage reset control register

SCULV_LVRCR is a 32-bit register, and available at 32/16/8-bit access.

SCULV_LVRCR=0x4000_5104

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			LVREN
-			0x00
-			RW

7 0	LVREN	LVR Enable bits. These bits are cleared to 0x00 by only POR and retained by other reset signals.	
		0x55	Disable low voltage reset.
		Others	Enable low voltage reset.

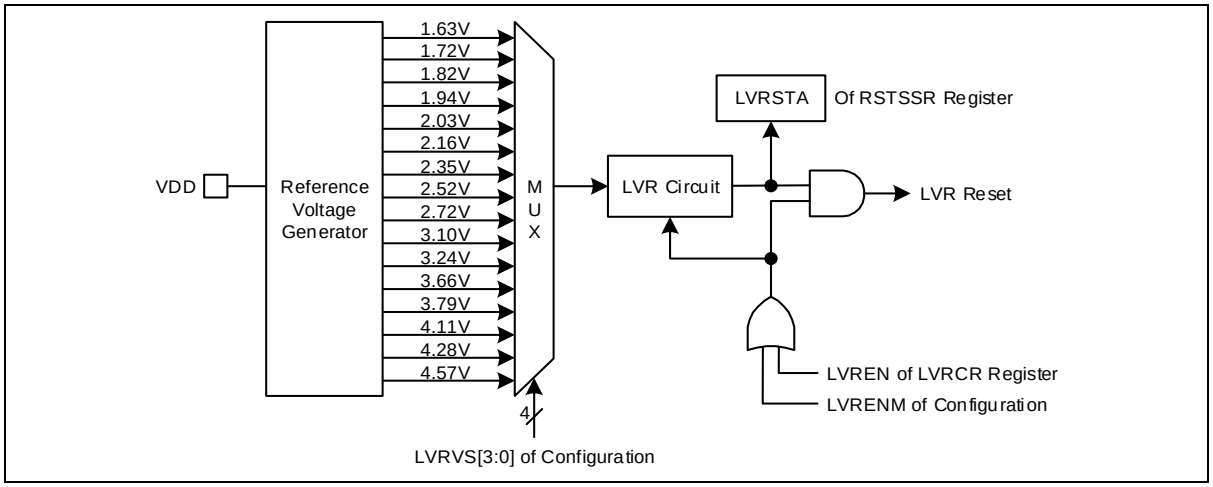


Figure 30. LVR Block Diagram

4.6.37 SCULV_LVRCNFIG: Configuration for low voltage reset

Low voltage indicator control register is a 32-bit register, and available at 32/16/8-bit access.

SCULV_LVRCNFIG =0x4000_5108

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
WTIDKY		LVRENM	Reserved	LVRVS
0x0000		0x00	-	0xF
WO		RW	-	RW

31 24	WTIDKY	Write Identification Key	
		On writes, write 0x72A5 to these bits, otherwise the write is ignored.	
15 8	LVRENM	LVR Reset Operation Control Master Configuration	
		0xAA	LVR operation is decided by the LVREN of LVRCR register
		Others	Master enable LVR operation
3 0	LVRVS	LVR Voltage Selection bits.	
		1111	1.63V
		1110	1.72V
		1101	1.82V
		1100	1.94V
		1011	2.03V
		1010	2.16V
		1001	2.35V
		1000	2.52V
		0111	2.72V
		0110	3.10V
		0101	3.24V
		0100	3.66V
		0011	3.79V
		0010	4.11V
		0001	4.28V
		0000	4.57V

5. PCU and GPIO

Port Control Unit (PCU) configures and controls external I/Os as listed in the followings:

- External signal directions of each pins
- Interrupt trigger mode for each pins
- Internal pull-up/down register control and open drain control

General Purpose Input/Output (GPIO) is corresponding to the most pins except dedicated-function pins. GPIO ports are controlled by GPIO block as listed in the followings:

- Output signal level (H/L) select
- External interrupt interface
- Pull up/down enable or disable

Four pins in Table 24 are assigned for PCU and GPIO blocks.

Table 24. PCU and GPIO pins

Pin name	Type	Description
PA	IO	PA0 to PA15
PB	IO	PB0 to PB15
PC	IO	PC0 to PC15
PF	IO	PF0 to PF2

5.1 PCU and GPIO Block diagram

Figure 31 describes PCU in block diagram.

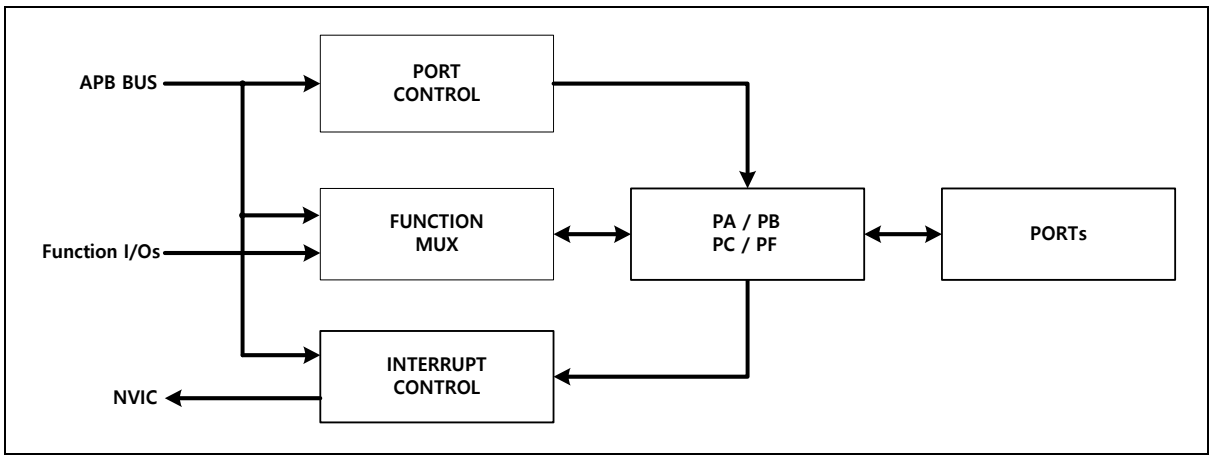


Figure 31. PCU Block Diagram

Figure 32 and Figure 33 describes GPIO in block diagram, and Figure 34 introduces external interrupt I/O pins.

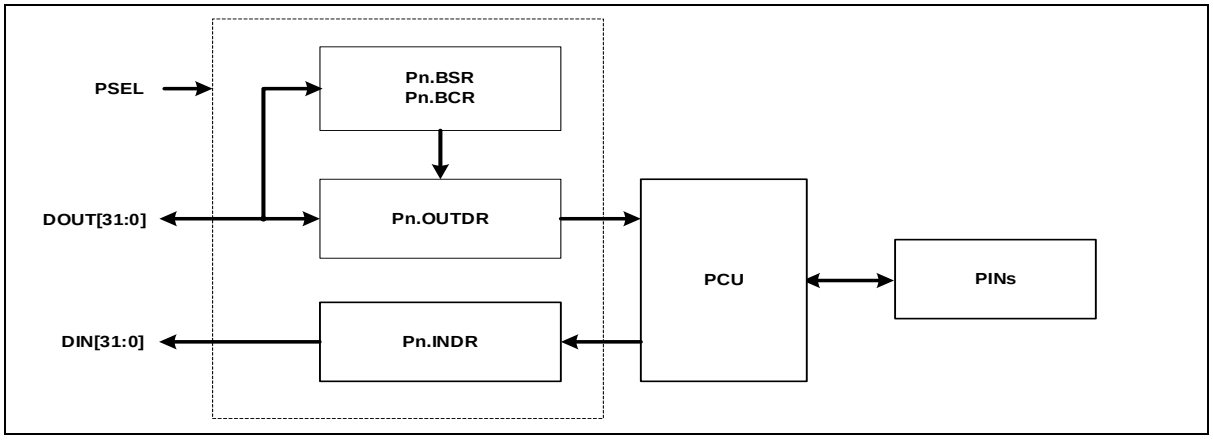


Figure 32. GPIO Block Diagram(Except PC13, PC14, PC15)

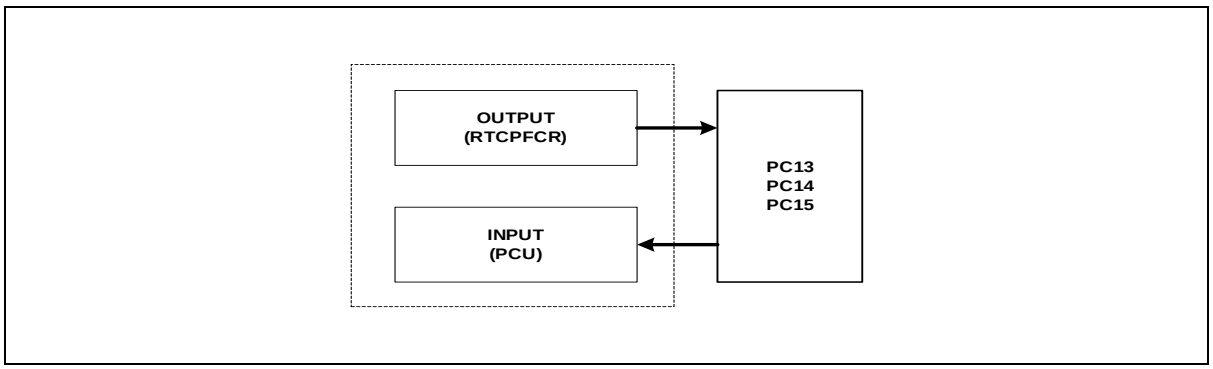


Figure 33. PC13, PC14, PC15 Block Diagram

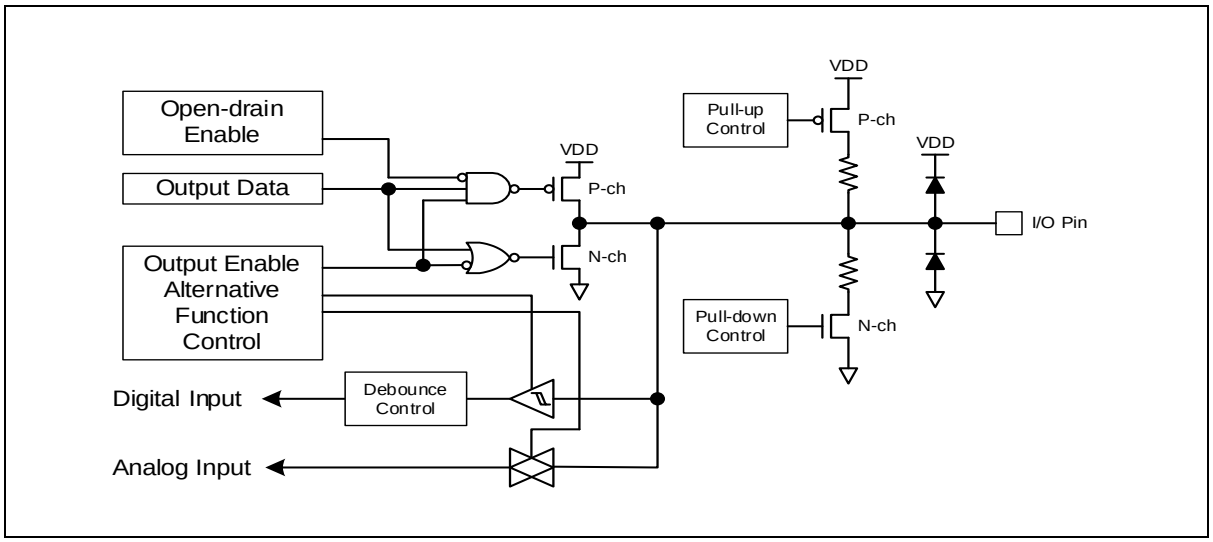


Figure 34. I/O Port Block Diagram (External Interrupt I/O Pins)

5.2 Pin multiplexing

GPIO pins have alternative function pins. Table 25 shows pin multiplexing information.

Table 25. GPIO Alternative Function

Pin name	Alternative function										
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10
PA0	USART13_TXD	USART13_MOSI		CP0_OUT	EC20			RTC_TAMP2			AN0 CP0_N2
PA1	USART13_RXD	USART13_MISO	USART11_SS		EC13	T13CAP	T13OUT	T20CAP	T20OUT		AN1 CP0_P0
PA2	USART11_TXD		USART11_MOSI	CP1_OUT	EC10	T10CAP	T10OUT	T20CAP	T20OUT		AN2 CP1_N2
PA3	USART11_RXD		USART11_MISO		EC10	T10CAP	T10OUT	T20CAP	T20OUT		AN3 CP1_P0
PA4	SPI20_SS		USART11_SCK								AN4 CP0_N0 CP1_N0 DAC_OUT1
PA5	SPI20_SCK				EC20						AN5 CP0_N1 CP1_N1 DAC_OUT2
PA6	SPI20_MISO	CP0_OUT		BLNK30	EC11	T11CAP	T11OUT	T40CAP_CH1	T40OUT	RD	AN6
PA7	SPI20_MOSI	CP1_OUT		PWM30AB	EC12	T12CAP	T12OUT	T40CAP_CH2	T40OUT	A19	AN7
PA8		USART10_SCK		PWM30AA						AD07	
PA9	USART10_TXD	USART10_MOSI		PWM30BA						AD06	
PA10	USART10_RXD	USART10_MISO		PWM30CA	EC21					AD05	
PA11	CP0_OUT									AD04	USBDM
PA12	CP1_OUT	USART10_SS		EC30	T30CAP					AD03	USBDP
PA13		SWDIO									
PA14	USART11_TXD	SWCLK	USART11_MOSI								
PA15	USART11_RXD	USART13_SS	USART11_MISO	SPI20_SS	EC21					ALE	
PB0		USART12_SCK		PWM30BB		T21CAP	T21OUT	T40CAP_CH3	T40OUT	A16	AN8
PB1		USART12_SS		PWM30CB	EC13	T13CAP	T13OUT			AD15	AN9
PB2					EC13	T13CAP	T13OUT			AD14	
PB3				SPI20_SCK		T20CAP	T20OUT			AD02	
PB4				SPI20_MISO				T40CAP_CH1	T40OUT	AD01	
PB5				SPI20_MOSI		T21CAP	T21OUT	T40CAP_CH2	T40OUT	AD00	
PB6	USART10_TXD	USART10_MOSI	I2C0_SCL		EC11	T11CAP	T11OUT	T40CAP_CH3	T40OUT	LWR	
PB7	USART10_RXD	USART10_MISO	I2C0_SDA		EC12	T12CAP	T12OUT			CS0	

Table 25. GPIO Alternative Function (continued)

Pin name	Alternative function										
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10
PB8		I2C0_SCL			EC11	T11CAP	T11OUT				
PB9		I2C0_SDA		SPI21_SS	EC12	T12CAP	T12OUT				
PB10	USART12_TXD	USART12_MOSI	I2C1_SCL	SPI21_SCK		T20CAP	T20OUT			AD13	
PB11	USART12_RXD	USART12_MISO	I2C1_SDA			T20CAP	T20OUT			AD12	
PB12	SPI21_SS	USART12_SCK		BLNK30	EC13	T13CAP	T13OUT			AD11	
PB13	SPI21_SCK		I2C1_SCL	PWM30AB						AD10	
PB14	SPI21_MISO	USART12_SS	I2C1_SDA	PWM30BB	EC10	T10CAP	T10OUT			AD09	
PB15	SPI21_MOSI			PWM30CB	EC10	T10CAP	T10OUT	T21CAP	T21OUT	AD08	
PC0										LWR	AN10 CP0_P1
PC1										AD00	AN11 CP1_P1
PC2	SPI21_MISO									AD01	AN12
PC3	SPI21_MOSI				EC13	T13CAP	T13OUT			AD02	AN13
PC4	USART12_TXD	USART12_MOSI						EC40		A18	AN14
PC5	USART12_RXD	USART12_MISO				T21CAP	T21OUT			A17	AN15
PC6						T21CAP	T21OUT	T40CAP_CH1	T40OUT	NWAIT	
PC7								T40CAP_CH2	T40OUT	UDS	
PC8								T40CAP_CH3	T40OUT	UWR	
PC9								EC40		CS3	
PC10	USART13_TXD	USART13_MOSI	USART12_TXD	USART12_MOSI						LDS	
PC11	USART13_RXD	USART13_MISO	USART12_RXD	USART12_MISO	EC21					RD	
PC12		USART13_SCK		USART12_SCK						CS2	
PC13								RTC_TAMP1			
PC14											SXIN
PC15											SXOUT
PF0		CLKO									XIN
PF1											XOUT
PF2				USART12_SS		T21CAP	T21OUT	EC40		CS1	

NOTES:

1. On connection with debugger host, SWCLK and SWDIO pins are always for SW-DP pins.
   So, the corresponding bits of PB_MOD/PB_TYP/PB_AFSR1/PB_PUPD registers may not be written by software.
2. RTCOUT must be set in PC13MODE in the RTCPFCR register.

5.3 Registers

Base address of PCU is introduced in the followings:

Table 26. Base Address of PCU

Name	Base address	Description
PA	0x4000_1000	General Port A
PB	0x4000_1100	General Port B
PC	0x4000_1200	General Port C
PF	0x4000_1500	General Port F

Table 27. PCU and GPIO Register Map

Name	Offset	Type	Description	Reset value	Reference
Pn_MOD	0x0000	RW	Port n Mode Register	0xXXXX_XXXX	5.3.1
Pn_TYP	0x0004	RW	Port n Output Type Selection Register	0x0000_0000	5.3.2
Pn_AFSR1	0x0008	RW	Port n Alternative Function Selection Register 1	0xXXXX_XXXX	5.3.3
Pn_AFSR2	0x000C	RW	Port n Alternative Function Selection Register 2	0x0000_0000	5.3.4
Pn_PUPD	0x0010	RW	Port n Pull-up/down Resistor Selection Register	0x0000_XXXX	5.3.5
Pn_INDR	0x0014	RO	Port n Input Data Register	0x0000_XXXX	5.3.6
Pn_OUTDR	0x0018	RW	Port n Output Data Register	0x0000_0000	5.3.7
Pn_BSR	0x001C	WO	Port n Output Bit Set Register	0x0000_0000	5.3.8
Pn_BCR	0x0020	WO	Port n Output Bit Clear Register	0x0000_0000	5.3.9
Pn_OUTDMSK	0x0024	RW	Port n Output Data Mask Register	0x0000_0000	5.3.10
Pn_DBCR	0x0028	RW	Port n Debounce Control Register	0x0000_0000	5.3.11
Pn_IER	0x002C	RW	Port n interrupt enable register	0x0000_0000	5.3.12
Pn_ISR	0x0030	RW	Port n interrupt status register	0x0000_0000	5.3.13
Pn_ICR	0x0034	RW	Port n interrupt control register	0x0000_0000	5.3.14
PCR.USBCON	0x0540	RW	USB Control Register	0x0000_0000	5.3.15
PCU_PORTEN	0x0FF0	WO	Port Access Enable	0x0000_0000	5.3.16

NOTE:

1. Where n = A, B, C, and F

5.3.1 Pn_MOD: PORT n mode register

Input or output control of each port pin. Each pin can be configured as an input pin, an output pin or an alternative function pin. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_MOD=0x4000_1000, PB_MOD=0x4000_1100
PC_MOD=0x4000_1200, PF_MOD=0x4000_1500

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
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8
00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	MODEx	Port n Mode Selection bits, x:0 to 15	
		00	Input mode
		01	Output mode
		10	Alternative function mode
		11	Reserved

NOTE :
For exception, the reset value is, PA_MOD : 0xEBFFFFFF / PB_MOD : 0xFFF0FFFF / PC_MOD : 0xFFFFFFFF / PF_MOD : 0x0000003F
PC13, PC14, PC15 output mode must use RTCPFCR register. Refer to 23.2.15 RTCPFCR: RTC pin function configuration register

5.3.2 Pn_TYP: Port n output type selection register

Pn_TYP selects control option from a Push-pull output and Open-drain output for each port pin. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_MOD=0x4000_1004, PB_MOD=0x4000_1104
PC_MOD=0x4000_1204, PF_MOD=0x4000_1504

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		TYP15	TYP14	TYP13	TYP12	TYP11	TYP10	TYP9	TYP8	TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	TYPx	Port n Output Type Selection bits, x:0 to 15	
		0	Push-pull output
		1	Open-drain output

NOTE : PC13, PC14, and PC15 cannot be used as open-drain. Only push-pull is available.

5.3.3 Pn_AFSR1: Port n alternative function selection register 1

Pn_AFSR1 registers must be set properly before using this port. Otherwise the port cannot be guaranteed for its functionality. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_AFSR1=0x4000_1008, PB_AFSR1=0x4000_1108
PC_AFSR1=0x4000_1208, PF_AFSR1=0x4000_1508

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
AFSB7	AFSB6	AFSB5	AFSB4	AFSB3	AFSB2	AFSB1	AFSB0
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3 4x	AFSBx	Port n Alternative Function Selection bits, x:0 to 7	
		0000	Alternative Function 0 (AF0)
		0001	Alternative Function 1 (AF1)
		0010	Alternative Function 2 (AF2)
		0011	Alternative Function 3 (AF3)
		0100	Alternative Function 4 (AF4)
		0101	Alternative Function 5 (AF5)
		0110	Alternative Function 6 (AF6)
		0111	Alternative Function 7 (AF7)
		1000	Alternative Function 8 (AF8)
		1001	Alternative Function 9 (AF9)
		1010	Alternative Function 10 (AF10)
		Others	Reserved

NOTE: When HSE is used as the system clock (MCLK), the AFSBx bits for PF0, PF1 must be configured as AF10 before changing the system clock and the value should not be changed.

5.3.4 Pn_AFSR2: Port n alternative function selection register 2

Pn_AFSR2 registers must be set properly before using this port. Otherwise the port cannot be guaranteed for its functionality. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_AFSR2=0x4000_100C, PB_AFSR2=0x4000_110C
PC_AFSR2=0x4000_120C, PF_AFSR2=0x4000_150C

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	
AFSB15	AFSB14	AFSB13	AFSB12	AFSB11	AFSB10	AFSB9	AFSB8
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4(x-8)+3 4(x-8)	AFSRx	Port n Alternative Function Selection bits, x:8 to 15	
		0000	Alternative Function 0 (AF0)
		0001	Alternative Function 1 (AF1)
		0010	Alternative Function 2 (AF2)
		0011	Alternative Function 3 (AF3)
		0100	Alternative Function 4 (AF4)
		0101	Alternative Function 5 (AF5)
		0110	Alternative Function 6 (AF6)
		0111	Alternative Function 7 (AF7)
		1000	Alternative Function 8 (AF8)
		1001	Alternative Function 9 (AF9)
		1010	Alternative Function 10 (AF10)
		Others	Reserved

NOTES:
When LSE is used as the system clock (MCLK) the AFSBx bits for PC14, PC15 must be configured as AF10 before changing the system clock and the value should not be changed.
For exception, the reset value of PA_AFSR2: 0x01100000.

5.3.5 Pn_PUPD: Port n pull-up/down resistor selection register

Each pin of the ports has on-chip pull-up/down resistor which can be configured by Pn_PUPD registers. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_PUPD=0x4000_1010, PB_PUPD=0x4000_1110
PC_PUPD=0x4000_1210, PF_PUPD=0x4000_1510

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
PUPD15	PUPD14	PUPD13	PUPD12	PUPD11	PUPD10	PUPD9	PUPD8	PUPD7	PUPD6	PUPD5	PUPD4	PUPD3	PUPD2	PUPD1	PUPD0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	PUPDx	Port n Pull-up/down Resistor Selection bits, x:0 to 15	
		00	Disable pull-up/down resistor
		01	Enable pull-up resistor
		10	Enable pull-down resistor
		11	Reserved

NOTES:
The pull-up resistors for the test mode pin (PB8, PB9) and the SWD pin (PA14, PA13) are initially enabled.
For exception, the reset value of PA_PUPD, PB_PUPD register is 0x14000000, 0x00050000 respectively.

5.3.6 Pn_INDR: Port n input data register

Each pin level status can be read in the Pn_INDR register. Even if a pin is alternative mode except analog mode and output in alternative mode, the pin level can be detected in the Pn_INDR register.

Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_INDR=0x4000_1014, PB_INDR=0x4000_1114
PC_INDR=0x4000_1214, PF_INDR=0x4000_1514

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		INDR15	INDR14	INDR13	INDR12	INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
-		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
-		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

x	INDRx	Port n Input Data bit, x:0 to 15

5.3.7 Pn_OUTDR: Port n output data register

Pn_OUTDR registers define output level of a pin when the pin is set as output and GPIO mode. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_OUTDR=0x4000_1018, PB_OUTDR=0x4000_1118
PC_OUTDR=0x4000_1218, PF_OUTDR=0x4000_1518

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		OUTDR15	OUTDR14	OUTDR13	OUTDR12	OUTDR11	OUTDR10	OUTDR9	OUTDR8	OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDRx	Port n Output Data bit, x:0 to 15 The OUTDR bits can be individually set/cleared by writing to the Pn_BSR/Pn_BCR register

NOTE: PC13, PC14, PC15 output mode must use RTCPFCR register. Refer to 23.2.15 RTCPFCR: RTC pin function configuration register

5.3.8 Pn_BSR: Port n output bit set register

Pn_BSR registers control each bit of Pn_OUTDR register. Writing '1' into the specific bit field will set a corresponding bit of Pn_OUTDR to '1'. Writing '0' has no effect.

Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_BSR=0x4000_101C, PB_BSR=0x4000_111C
PC_BSR=0x4000_121C, PF_BSR=0x4000_151C

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		BSR15	BSR14	BSR13	BSR12	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BSRx	Port n Output Set bit, x: 0 to 15. These bits are always read to 0x00	
		0	No effect
		1	Set the corresponding OUTDRx bit (automatically cleared to 0)

NOTE: PC13, PC14, PC15 output mode must use RTCPFCR register. Refer to 23.2.15 RTCPFCR: RTC pin function configuration register

5.3.9 Pn_BCR: Port n output bit clear register

Pn_BCR registers control each bit of Pn_OUTDR register. Writing '1' into the specific bit field will set a corresponding bit of Pn_OUTDR to '0'. Writing '0' has no effect.

Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_BCR=0x4000_1020, PB_BCR=0x4000_1120
PC_BCR=0x4000_1220, PF_BCR=0x4000_1520

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		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BCRx	Port n Output Clear bit, x: 0 to 15. These bits are always read to 0x00	
		0	No effect
		1	Clear the corresponding OUTDRx bit (automatically cleared to 0)

NOTE: PC13, PC14, PC15 output mode must use RTCPFCR register. Refer to 23.2.15 RTCPFCR: RTC pin function configuration register

5.3.10 Pn_OUTDMSK: Port n output data mask register

Pn_OUTDMSK registers protect each bit of Pn_OUTDR registers. Writing '1' into the specific bit field will protect a corresponding bit of Pn_OUTDR. Writing '0' is unmasked.

Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_OUTDMSK=0x4000_1024, PB_OUTDMSK=0x4000_1124
PC_OUTDMSK=0x4000_1224, PF_OUTDMSK=0x4000_1524

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		OUTDMSK15	OUTDMSK14	OUTDMSK13	OUTDMSK12	OUTDMSK11	OUTDMSK10	OUTDMSK9	OUTDMSK8	OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDMSKx	Port n Output Data Mask bit, x: 0 to 15.	
		0	Unmask. The corresponding OUTDR bit can be changed.
		1	Mask. The corresponding OUTDRx bit is protected.

NOTE: PC13, PC14, PC15 output mode must use RTCPFCR register. Refer to 23.2.15 RTCPFCR: RTC pin function configuration register

5.3.11 Pn_DBCR: Port n debounce control register

Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A, B, C, and F).

PA_DBCR=0x4000_1028, PB_DBCR=0x4000_1128
PC_DBCR=0x4000_1228, PF_DBCR=0x4000_1528

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		DBEN15	DBEN14	DBEN13	DBEN12	DBEN11	DBEN10	DBEN9	DBEN8	DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	DBENx	Port n Debounce Enable bit, x: 0 to 15. Port debounce length = selected debounce clock period * (4 to 5)	
		0	Disable debounce filter
		1	Enable debounce filter

NOTES:
If a level is not detected on an enabled pin three or more times in a row at the sampling clock, the signal is eliminated as noise.
The port debounce should be disabled before Power Down mode.

5.3.12 Pn_IER: Port n interrupt enable register

Each pin of A31G32x can be an external interrupt source. In this case, both of edge trigger interrupt and level trigger interrupt are supported.

Pn_IER registers can configure the interrupt mode (n = A, B, C, and F).

PA_IER=0x4000_102C, PB_IER=0x4000_112C
PC_IER=0x4000_122C, PF_IER=0x4000_152C

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
PIE15 PIE14 PIE13 PIE12	PIE11 PIE10 PIE9 PIE8	PIE7 PIE6 PIE5 PIE4	PIE3 PIE2 PIE1 PIE0
00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
RW RW RW RW	RW RW RW RW	RW RW RW RW	RW RW RW RW

2x+1 2x	PIEx	Port n Pin interrupt Enable Selection bits, x:0 to 15	
		00	Disable Interrupt
		01	Enable interrupt as level trigger mode
		10	Reserved
		11	Enable interrupt as edge trigger mode

5.3.13 Pn_ISR: Port n interrupt status register

When an interrupt is delivered to CPU, the interrupt status can be detected by reading Pn_ISR registers. Pn_ISR registers will report a source pin of the interrupt and a type of the interrupt (n = A, B, C, and F).

PA_ISR=0x4000_1030, PB_ISR=0x4000_1130
PC_ISR=0x4000_1230, PF_ISR=0x4000_1530

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			
PIS15	PIS14	PIS13	PIS12	PIS11	PIS10	PIS9	PIS8	PIS7	PIS6	PIS5	PIS4	PIS3	PIS2	PIS1	PIS0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	PISx	Port n Pin interrupt Status bits, x:0 to 15	
		00	No interrupt event
		01	Low level interrupt or Falling edge interrupt event is present.
		10	High level interrupt or Ralling edge interrupt event is present.
		11	Both of rising and falling interrupt event is present in edge trigger interrupt mode. Not available in level trigger interrupt mode.

5.3.14 Pn_ICR: Port n interrupt control register

Pn_ICR registers control interrupt mode of port pins (n = A, B, C, and F).

PA_ICR=0x4000_1034, PB_ICR=0x4000_1134
PC_ICR=0x4000_1234, PF_ICR=0x4000_1534

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			
PIC15	PIC14	PIC13	PIC12	PIC11	PIC10	PIC9	PIC8	PIC7	PIC6	PIC5	PIC4	PIC3	PIC2	PIC1	PIC0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	PICx	Port n Pin interrupt Control bits, x:0 to 15	
		00	Prohibit external interrupt
		01	Low level interrupt or Falling edge interrupt mode
		10	High level interrupt or Ralling edge interrupt mode
		11	Both of rising and falling edge interrupt mode Not support for level trigger interrupt mode.

5.3.15 PCU_USBCON: USB control register

PCU_USBCON register controls USB pull-up/pull-down, VIL, VIH (only PA11, PA12).

USBCON=0x4000_1540

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	TX_DNOPT	Reserved	TX_UPOPT	CONTROL_EN	PHY_SUSPND	VREF_SEL	Reserved	PD_ENDN	PD_ENDP	PU_ENDN	PU_ENDP	Reserved	VIH_CON	Reserved	VIH_CON	Reserved	PD_CON	Reserved	PU_CON
-	000	-	000	0	0	0	-	0	0	0	0	-	000	-	000	-	000	-	000
-	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	-	RW	-	RW	-	RW	-	RW

30 28	TX_DNOPT	TX Driver Up Driving Strength Option			
		000	+ 0%	100	+ 15%
		001	+ 5%	101	+ 20%
		010	+ 10%	110	+ 25%
		011	-	111	+ 30%
26 24	TX_UPOPT	TX Driver Down Driving Strength Option			
		000	+ 0%	100	+ 15%
		001	+ 5%	101	+ 20%
		010	+ 10%	110	+ 25%
		011	-	111	+ 30%
23	CONTROL_EN	USB PHY pull up, pull down, VIL, VIH control enable			
22	PHY_SUSPND	PHY SUSPEND signal (active HIGH)			
21	VREF_SEL	Single ended Receiver hysteresis reference Voltage Select			
		0	BGR 1.2V		
		1	VDD15 1.5V		
19	PD_ENDN	DN pin Pull down enable			
18	PD_ENDP	DP pin Pull down enable			
17	PU_ENDN	DN pin Pull up enable			
16	PU_ENDP	DP pin Pull up enable			
14 12	VIH_CON	VIH level select register			
		000	0.4V	100	0.8V
		001	0.5V	101	0.9V
		010	0.6V	110	1.0V
		011	0.7V	111	1.1V

10 8	VIL_CON	VIL level select register			
		000	1.6V	100	2.0V
		001	1.7V	101	2.1V
		010	1.8V	110	2.2V
		011	1.9V	111	2.3V
6 4	PD_CON	Pull down resister select register			
		000	17.5KOhm	100	14.5KOhm
		001	16.3KOhm	101	13.8KOhm
		010	16KOhm	110	13KOhm
		011	15.3KOhm	111	12.3KOhm
2 0	PU_CON	Pull up resister select register			
		000	1.97KOhm	100	1.26KOhm
		001	1.8KOhm	101	1.1KOhm
		010	1.63KOhm	110	
		011	1.46KOhm	111	

5.3.16 PCU_PORTEN: Port access enable

PCU_PORTEN register enables the register writing permission of all PCU registers.

PCU_PORTEN=0x4000_1FF0

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			PORTEN
-			--
-			WO

7 0	PORTEN	Writing the sequence of 0x15 and 0x51 in this register enables writing to PCU registers, and writing other values protects all PCU registers from writing.

NOTE: Refer to the followings to use PORTEN;

```
PORTEN=0x15; PORTEN=0x51;                                   // enable PORTEN
…                                                           // set Pn_MOD,
Pn_TYP, Pn_AFSR1,2, Pn_PUPD. Pn_DBCR, Pn_IER, Pn_ICR
PORTEN=0;
                                                            // disable PORTEN
```

5.4 Functional description

If an input function of a certain I/O port is used by Pin Control Register, an output function of the I/O port is disabled. Function of each port can be different in accordance with an Alternative Function Selection Register.

Input Data Register captures current data of the I/O pin or debounced input data at every GPIO clock cycle.

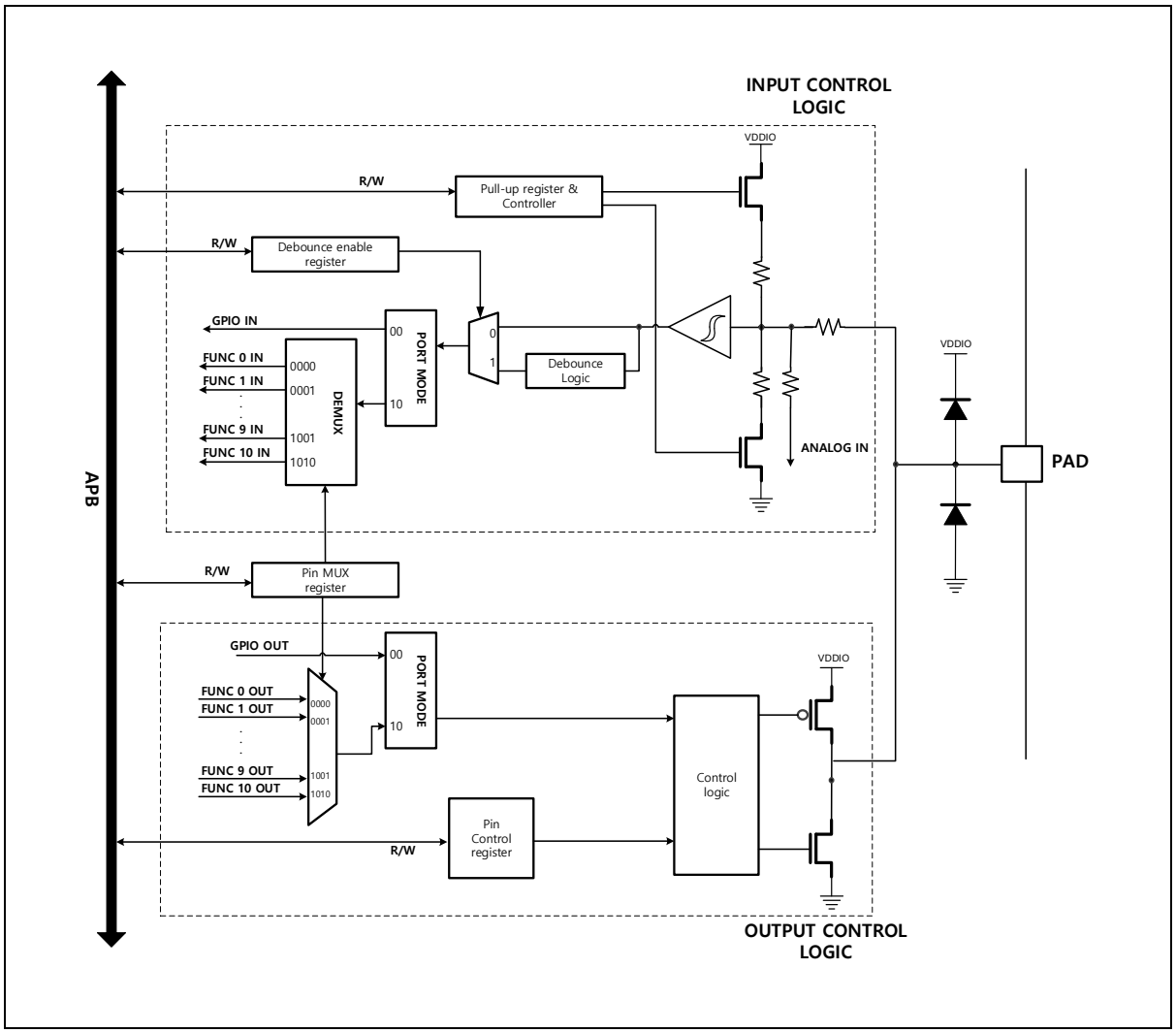


Figure 35. Port Diagram

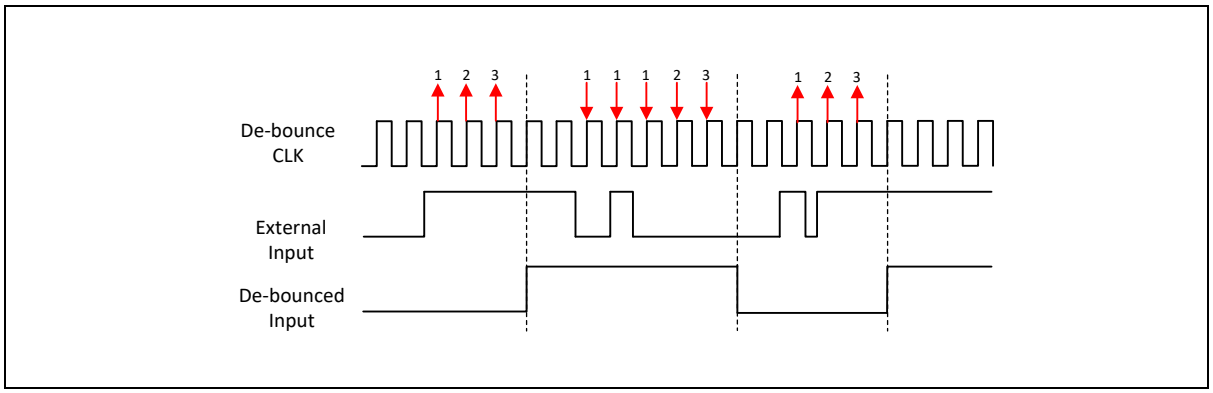


Figure 36. Debounce Function Timing Diagram

- When configured as output, the value written to the GPIO Output Data Register is output on the I/O Pin.
- When setting the Bit Set Register, GPIO Output Data Register set the high.
- When setting the Bit Clr Register, GPIO Output Data Register set the Low.
- The Input Data Register captures the data present on the I/O pin or Debounced input data at every GPIO Clock cycle.

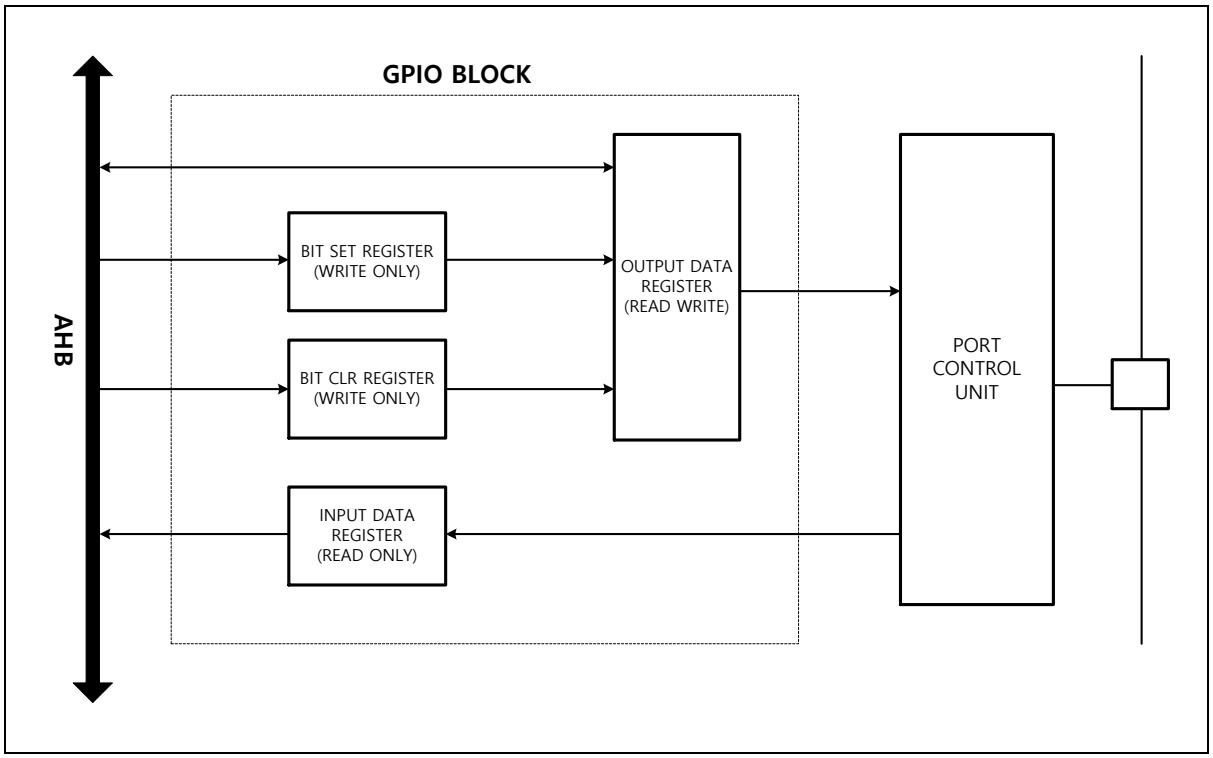


Figure 37. GPIO Diagram

6. Flash memory controller

Flash Memory Controller is an internal flash memory interface controller, and includes following features as shown below:

- 128 or 64KB Flash code memory
- Wait, 1-wait, 2-wait(default)
- Read protection support
- Self-Program support
- User option area
- 3-page (each 512 Bytes)
- Erase, Program in user mode

Table 28. Flash Memory Controller Features

Item	Description	
Size	64KB	128KB
Start Address	0x0000_0000	0x0000_0000
End Address	0x0001_0000	0x0002_0000
Page Size	512-byte	512-byte
Total Page Count	128 pages	256 pages
PGM Unit	32-bit	32-bit
Erase Unit	512-byte/ 1KB/ 4KB/ bulk	512-byte/ 1KB/ 4KB/ bulk

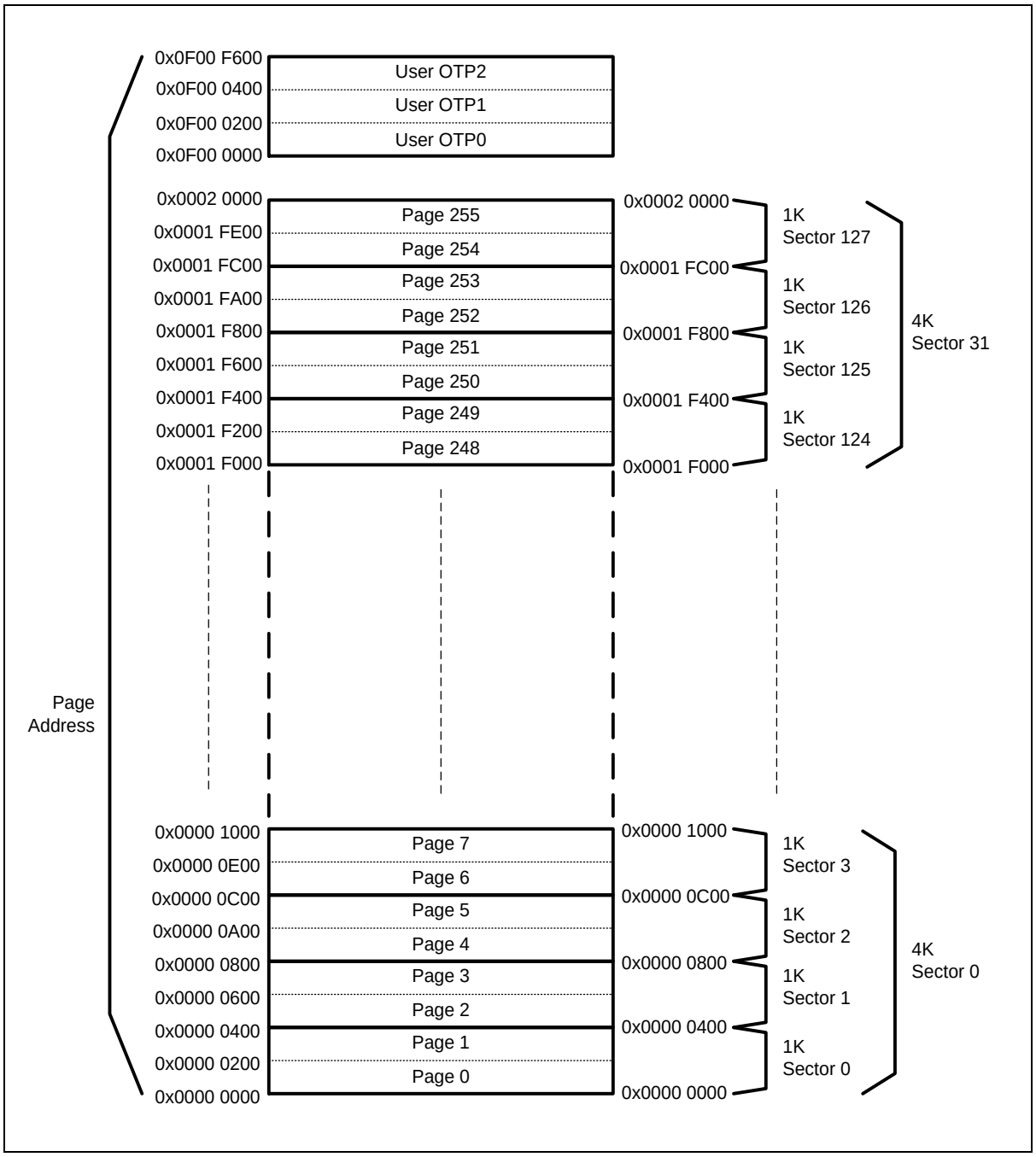


Figure 38. Flash Memory Map (128 KB Code Flash)

6.1 Registers

Base address of flash memory controller is introduced in the followings:

Table 29. Base Address of Flash Memory Controller

Name	Base address
Flash controller	0x4000_0100

Table 30. FMC Register Map

Name	Offset	Type	Description	Reset value	Reference
FMC_MR	0x0004	R/W	Flash Memory Mode Select Register	0x0100_0000	6.1.1
FMC_CR	0x0008	R/W	Flash Memory Control Register	0x0000_0000	6.1.2
FMC_AR	0x000C	R/W	Flash Memory Address Register	0x0000_0000	6.1.3
FMC_DR	0x0010	R/W	Flash Memory Data Register	0x0000_0000	6.1.4
FMC_BUSY	0x0018	R/W	Flash Write Busy Status Register	0x0000_0000	6.1.5
FMC_CRC	0x0020	R/W	Flash CRC16 check value	0x0000_FFFF	6.1.6
FMC_CFG	0x0030	R/W	Flash Memory Configuration Register	0x0000_8200	6.1.7
FMC_WPROT	0x0034	R/W	Write Protection Register	0xFFFF_FFFF	6.1.8
FMC_LOCK	0x003C	R/W	Flash LOCK Register	0x0000_00FF	6.1.9

6.1.1 FMC_MR: Flash memory mode register

FMC_MR is an internal flash memory mode register. Size of this register is 32-bit.

Internal flash memory mode register. This register is 32-bit register.

FMC_MR=0x4000_0104

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			ACODE
-			0x00
.			RW

7 0	ACODE	5A → A5	Flash mode entry
		A5 → 5A	Trim mode entry
		81 → 28	AMBA mode entry
		66 → 99	PROT mode entry

6.1.2 FMC_CR: Flash memory control register

FMC_CR is an internal flash memory control register.

FMC_CR=0x4000_0108

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	LOCKSEL	SELFPGM	Reserved		IFEN	Reserved	BBLOCK	MAS	SECT4K	SECT1K	PMODE	WADCK	PGM	ERS	HVEN
-	0	0	-		0	-	0	0	0	0	0	0	0	0	0
.	RW	RW	.		RW	.	RW	RW	RW	RW	RW	RW	RW	RW	RW

24	LOCKSEL		LOCK (read protection) access enable.
23	SELFPGM		When this bit is set ("1"), PGM/ERS/HVEN will be cleared automatically after WRBUSY falling edge. It also enable CPU wait control when HVEN bit is set(1) (start of program or erase operation) It also affects to PMODE bit operation.
12	IFEN	0	Info(OTP1/2/3) block enable
		1	OTP1/2/3 area enable (it works with OTP3EN to OTP1EN) for PMODE operation
8	BBLOCK	0	Boot Block (1st 4KB) not protected from Mass(Bulk) Erase
		1	Boot Block (1st 4KB) protection enable from Mass(bulk) erase
7	MAS	0	Mass (bulk) erase disable
		1	Mass (bulk) erase enable.
6	SECT4K	0	Sector 4K erase disable
		1	Sector 4K erase enable
5	SECT1K	0	Sector 1K erase disable
		1	Sector 1K erase enable

4	PMODE	0	Normal mode
		1	PMODE enable(Flash Address path is connected with FMAR) PMODE only valid when SELFPGM bit was not set(when SELFPGM = 0)
3	WADCK	0	Program/Erase address data latch clock disable
		1	Program/Erase address data latch clock enable, this bit assert for one system clock period so user cannot read
2	PGM	0	Program mode disable
		1	Program mode enable
1	ERS	0	Erase mode disable
		1	Erase mode enable
0	HVEN	0	High Voltage cycle disable
		1	High Voltage cycle enable (start program or erase cycle) User must set and clear in PMODE. In SELFPGM mode, user must set HVEN then HVEN will be cleared automatically after WRBUSY goes low.

6.1.3 FMC_AR: Flash memory address register

FMC_AR is an internal flash memory program/ erase/ address register.

FMC_AR=0x4000_010C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FADDR															
0x0000															
RW															

15 0	FADDR	Word (32-bit) base address: 64K-word address for 256KB Flash. Auto Incremental after WADCK trigger (after latching of target address).

6.1.4 FMC_DR: Flash data input register

FMC_DR is an internal flash memory data input register.

FMC_DR=0x4000_0110

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
FDATA																															
0x00000000																															
RW																															

31 0	FDATA	Word size(32-bit)

6.1.5 FMC_BUSY: Flash write busy status register

FMC_BUSY is a flash write (program/erase) busy status monitor register. This register is a 1-bit read only register.

FMC_BUSY=0x4000_0118

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	WRBUSY
-	0
-	RO

0	WRBUSY	Write Busy status bit FLBUSY bit goes high after set HVEN bit (in CTRL register). FLBUSY bit goes low when WRBUSY becomes low after program (or erase) complete.

6.1.6 FMC_CRC: Flash CRC check register

FMC_CRC is the built-in CRC calculates the flash data automatically.

- 16-bit read only register [15:0], which enabled by CRCEN bit of CFG register
- At least 16-word read to get a CRC value

FMC_CRC=0x4000_0120

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		CRC-CCITT	
-		0xFFFF	
-		RW	

15 0	CRC- CCITT	CRC- CCITT check value read register polynomial: (1 + x5 + x12 + x16) data width: 32 (the first serial bit is D[31])

6.1.7 FMC_CFG: Flash memory configuration register

FMC_CFG is an internal flash memory Configuration register.

FMC_CFG=0x4000_0130

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
WTIDKY		Reserved	WAIT	CRCINIT	CRCEN	Reserved
0x7858		-	1 0	0	0	-
WO		-	RW	RW	RW	-

31 16	WTIDKY		Write Identification Key. On writes, write 0x7858 to these bits, otherwise the write is ignored.
9 8	WAIT		This bits only be written in AMBA mode and MSB 16-bit (bit [31:16]) must be 0x7858
		00	WAIT is 00, flash access in 1 cycle (0-wait)
		01	WAIT is 01, flash access in 2 cycles (1-wait)
		10	WAIT is 10, flash access in 3 cycles (2-wait) – default
		11	WAIT is 11, flash access in 4 cycles (3-wait)
7	CRCINIT	0	When this bit is set('1'), CRC register will be initialized It should be reset again before read flash to generate CRC16 calculation (Initial value of FMC_CRC is 0xFFFF)
6	CRCEN	0	CRC16 enable CRC value will be calculated at every flash read timing

6.1.8 FMC_WPROT: Write protection register

FMC_WPROT is an internal flash memory write protection register.

FMC_WPROT=0x4000_0134

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
WPROT[31:0]			
0xFFFF_FFFF			
RW			

31 0	WPROT	Write protection Each 4 KB segments for whole memory address (needs 32 bits for 128KB flash)

NOTES:

1. Each bit individually protects the 4KB area (Bit0: 0~0xFFF, Bit1: 0x1000~0x1FFF).
2. The FM_WPROT register can only be modified in PROT (FM_MR = 66-> 99) mode.

6.1.9 FMC_LOCK: Flash lock register

FMC_LOCK is an internal flash memory read protection register.

FMC_LOCK=0x4000_013C

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			RPROT
-			0x0000_00FF
-			RW

7 0	RPROT	Read protection 0x0000_00FF is Default. Any other value will lock flash(enable read protection) In user mode, 0xFF cannot be written To unlock, user must erase LOCK area of Flash next to MAS(bulk) erase - 1st MAS(bulk) erase and then erase LOCK area - To unlock the flash, pin reset or power on reset required - When read protection occurs, Registers are shown as 0xAA55AA55.

NOTE:

1. The FM_WPROT register can only be modified in PROT. (FM_MR = 66-> 99) mode.

6.2 Functional description

6.2.1 Flash erase and program examples

Basic steps of flash memory programming consist of the followings. Minimum Program or Erase unit is a Page, and 32-word (5-byte) becomes a page.

- Page erase
- Page program
- Self page erase
- Self page program

For all of erase operations, pre-program operation is required to prevent over erase of flash memory cells. In addition, a user must enable 48MHz internal oscillator first to erase or program flash.

Erase example

1. Enable flash mode to write FMC_MCR register (write 0x5A and then write 0xA5 into FMC_MR).
2. Set target Page address in FMC_AR.
3. Set PMODE bit first.
4. Set ERS, WADCK, HVEN bits of FMC_MCR.
5. Wait until IDLE bit of FMC_MR register becomes "0" after erase.
6. Clear ERS, HVEN bits of FMC_CR.
7. Set 0x80 to FMC_BUSY.
8. Clear FMC_CR.
9. Clear Flash mode (write 0x00 and then write 0x00 into FMC_MR).

Program example

1. Enable flash mode to write FMC_CR register (write 0x5A and then write 0xA5 into FMC_MR).
2. Set PMODE bit first.
3. Set target Page address in FMC_AR.
4. Set PGM bits of FMC_CR.
5. Write word (32-bit) data into FMC_DR, address increased automatically based on word address.
6. Set WADCK, HVEN bits of FMC_CR.
7. Wait until IDLE bit of FMC_MR register becomes "0" after program.
8. Clear HVEN bits of FMC_CR.
9. Set 0x80 to FMC_ BUSY.
10. Clear PGM bits of FMC_CR.
11. Clear FMC_CR.
12. Clear Flash mode (write 0x00 and then write 0x00 into FMC_MR).

Self Erase example

1. Enable flash mode to write FMC_CR register (write 0x5A and then write 0xA5 into FMC_MR).
2. Set SELFPGM, ERS bits of FMC_CR.
3. Wait 5 clocks.
4. Write "0xFFFFFFFF" into target address.
5. Wait 5 clocks.
6. Clear ERS bits of FMC_CR.
7. Clear FMC_CR.
8. Wait 5 clocks.
9. Clear Flash mode (write 0x00 and then write 0x00 into FMC_MR).

Self Program example

1. Enable flash mode to write FMC_CR register (write 0x5A and then write 0xA5 into FMC_MR).
2. Set SELFPGM, ERS bits of FMC_CR.
3. Wait 5 clocks.
4. Write word (32-bit) data into target address.
5. Wait 5 clocks.
6. Clear ERS bits of FMC_CR.
7. Clear FMC_CR.
8. Wait 5 clocks.
9. Clear Flash mode (write 0x00 and then write 0x00 into FMC_MR).

7. Direct Memory Access Controller (DMAC)

The direct memory access (DMA) controller is used for high-speed data transfers between peripherals and memories. DMA enables quick data transfers having memory to memory copying or moving of data within memory.

- 4 channels
- Single transfer only
- Support 8/16/32-bit data size
- Support multiple buffer with same size
- Interrupt condition is transferred through a peripheral interrupt.

7.1 Block diagram

In this section, DMAC block diagram is introduced in Figure 39.

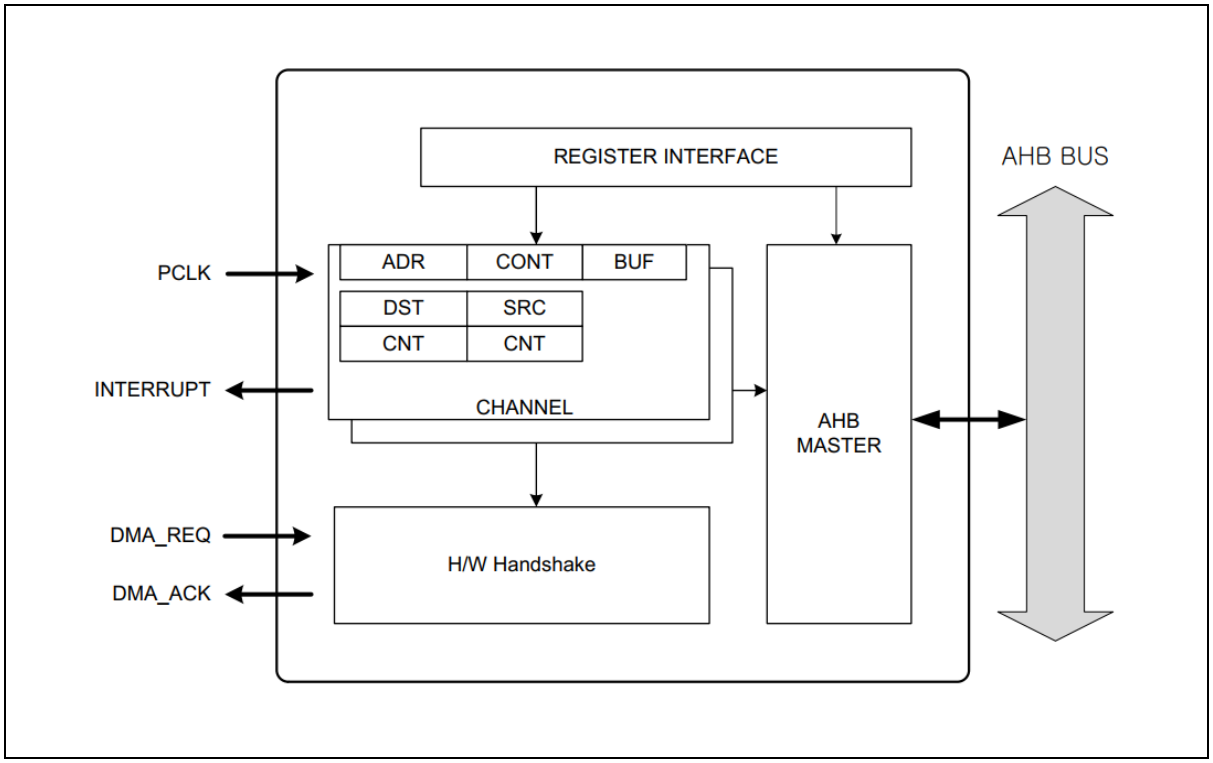


Figure 39. DMAC Block Diagram

7.2 Registers

Base address of DMAC is introduced in the followings:

Table 31. Base Address of DMAC

Name	Base address
DMACH0	0x4000_0400
DMACH1	0x4000_0410
DMACH2	0x4000_0420
DMACH3	0x4000_0430

Table 32. DMAC Register Map

Name	Offset	Type	Description	Reset value	Reference
DCn.CR	0x0000	RW	DMA Channel n Control Register	0x0000_0000	7.2.1
DCn.SR	0x0004	RW	DMA Channel n Status Register	0x0000_0000	7.2.2
DCn.PAR	0x0008	R	DMA Channel n Peripheral Address Register	0x0000_0000	7.2.3
DCn.MAR	0x000C	RW	DMA Channel n Memory Address Register	0x2000_0000	7.2.4

7.2.1 DCn.CR: DMA controller configuration register

DCn.CR registers are DMA operation control registers, and the register size is 32-bit.

DC0.CR=0x4000_0400 , DC1.CR=0x4000_0410
DC2.CR=0x4000_0420 , DC3.CR=0x4000_0430

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	TRANSCNT	Reserved	PERISEL	Reserved	SIZE	DIR	Reserved
-	0x000	-	0	-	00	0	-
-	RW	-	RW	-	RW	RW	-

27 16	TRANSCNT	Number of DMA transfer remained Required transfer number should be written before enable DMA transfer.	
		0	DMA transfer is done.
		N	N transfers are remained
11 8	PERISEL	Peripheral selection	
		N	Associated peripheral selection. Refer to DMA Peripheral connection table
3 2	SIZE	Bus transfer size.	
		00	DMA transfer is byte size transfer
		01	DMA transfer is half word size transfer
		10	DMA transfer is word size transfer
		11	Reserved
1	DIR	Select transfer direction.	
		0	Transfer direction is from memory to peripheral. (TX)
		1	Transfer direction is from peripheral to memory (RX)

NOTE: A DMA channel will be connected with selected peripheral. Below table shows peripheral selection numbers. This PERISEL field should be set with proper number of peripheral which will be connected with DMA interface.

Table 33. DMAC PERISEL Selection

PERISEL[3:0]	Associate peripheral
0	CHANNEL IDLE
1	SPI20 RX
2	SPI20 TX
3	SPI21 RX
4	SPI21 TX
5	USART10 RX
6	USART10 TX
7	USART11 RX
8	USART11 TX
9	USART12 RX
10	USART12 TX
11	USART13 RX
12	USART13 TX
13	CRC
14	ADC
15	Reserved

PERISEL cannot have the same value in different channels. If the same PERISEL value is written in more than one channel, proper operation cannot be guaranteed. Unused channel should have CHANNEL IDLE value in PERISEL bit positions.

7.2.2 DCn.SR: DMA controller status register

DCn.SR registers represent current status of DMA Controller, and enable DMA function. The register size is 8-bit.

DC0.SR=0x4000_0404 , DC1.SR=0x4000_0414
DC2.SR=0x4000_0424 , DC3.SR=0x4000_0434

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	EOT	Reserved	DMAEN
-	1	-	0
-	RO	-	RW

7	EOT	End of transfer.	
		0	Data to be transferred is existing. TRANSCNT shows non zero value
		1	All data is transferred. TRANSCNT shows now 0
0	DMAEN	DMA Enable	
		0	DMA is in stop or hold state
		1	DMA is running or enabled

7.2.3 DCn.PAR: DMA controller peripheral address register

DCn.PAR registers represent peripheral addresses.

DC0.PAR=0x4000_0408 , DC1.PAR=0x4000_0418
DC2.PAR=0x4000_0428 , DC3.PAR=0x4000_0438

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
Peripheral Base address Offset	PAR
0x4000	0x0000
RO	RW

15 0	PAR	Target Peripheral address of transmit buffer or receive buffer. User must set exact target peripheral buffer address in this field. If DIR is "0" this address is destination address of data transfer. If DIR is "1", this address is source address of data transfer.

7.2.4 DCn.MAR: DMA controller memory address register

DCn.MAR registers represent the memory addresses.

DC0.MAR=0x4000_040C , DC1.MAR=0x4000_041C
DC2.MAR=0x4000_042C , DC3.MAR=0x4000_043C

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
Memory Base address Offset		MAR	
0x2000		0x0000	
RO		RW	

15 0	MAR	Target memory address of data transfer. Address is automatically incremented according to SIZE bits when each transfer is done. If DIR is "0" this address is source address of data transfer. If DIR is "1", this address is destination address of data transfer.

7.3 Functional description

A DMA controller performs direct memory transfer by sharing the system bus with CPU core. The system bus is shared by two AHB (Advanced High-performance Bus) masters following the round-robin priority strategy. So the DMA controller can share the half of system bandwidth.

The DMA controller can be triggered only by a peripheral request. When a peripheral requests a transfer to the DMA controller, a corresponding channel is activated and the bus is accessed to transfer the requested data from memory to peripheral data buffer or vice versa.

Basic steps to trigger DMAC data transfer consist of following 10 steps:

1. Set both of peripheral address and memory address.
2. Configure DMA operation mode and transfer count.
3. Enable a DMA channel.
4. DMA request is occurred from the peripheral.
5. DMA activates the channel which was requested.
6. DMA reads data from source address and saves in internal buffer.
7. DMA writes the buffered data to destination address.
8. Transfer count number is decreased by '1'.
9. When the transfer count is '0', EOT flag is set and noticed to peripheral to issue the interrupt
10. DMA does not have interrupt sources, and the interrupt related DMA status can be shown from assigned peripheral interrupt.

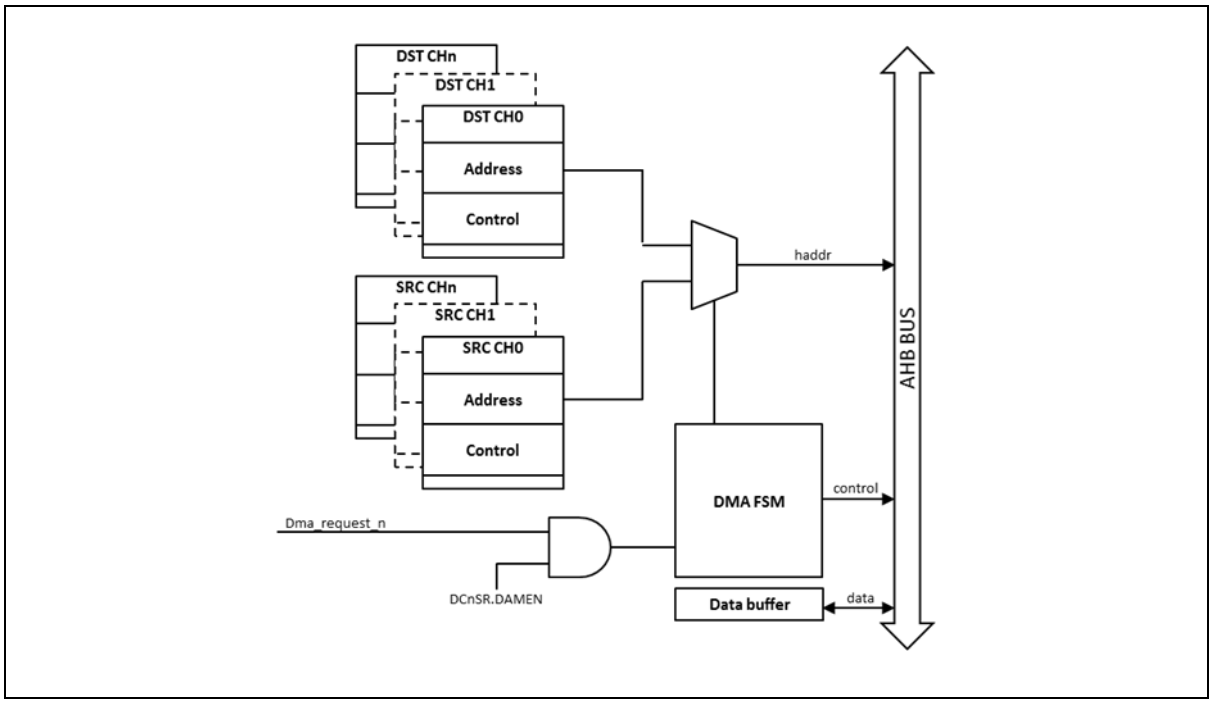


Figure 40. DMAC Functional Block Diagram

7.3.1 DMA operation

A user can start DMA operation by following the procedure introduced below:

1. Set DCn.CR registers of DMA.
   - [27:16]: Set a number of data to transfer to DMA.
   - [11:8]: Select a peripheral to connect with DMA.
   - [3:2]: Select a buffer size to transfer.
   - [1]: Set a transfer type from TX and RX.
2. Set MAR register of DMA (memory address to which DMA accesses).
3. Set PAR register of DMA (peripheral address to which DMA accesses).
4. Check the EOT flag of DMA start and DCn.SR register.
5. Check the DMA flag through status register of each peripheral.
6. DMA stops.

A sequence of DMAC operation is described in Figure 41.

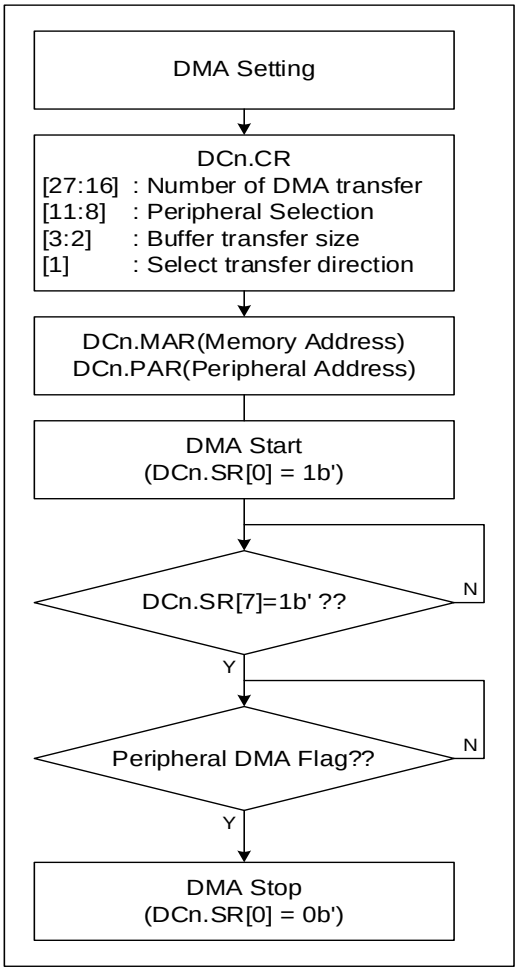


Figure 41. DMAC Operation Sequence

Figure 42 shows the functional timing diagram of DMAC. Transfer request from a certain peripheral is pended internally and it will invoke source data read transfer on the AHB bus. The read data from the source address is stored in the internal buffer. Then this data will be transferred to the destination address when the AHB bus is available.

Figure 42 introduces the timing diagram for a DMA transfer from peripheral to memory. There is 4-clock cycle latencies during accessing the peripheral. If the bus is occupied by different bus master, there are amount of bus waiting cycles.

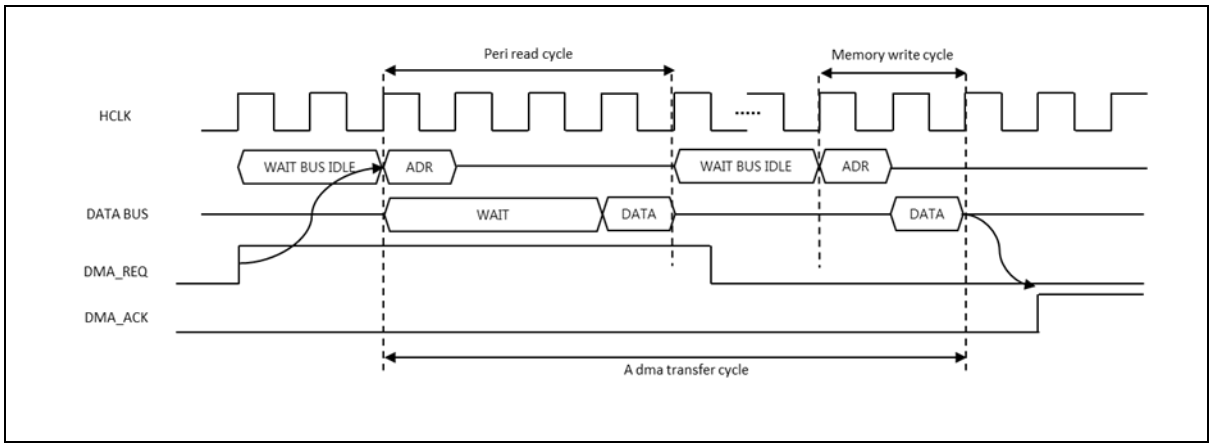


Figure 42. Timing Diagram of DMAC Transfer from Peripheral to Memory

Figure 43 introduces the timing diagram for a DMA transfer from memory to peripheral. There is 4-clock cycle latencies during accessing the peripheral. If the bus is occupied by different bus master, there are amount of bus waiting cycles.

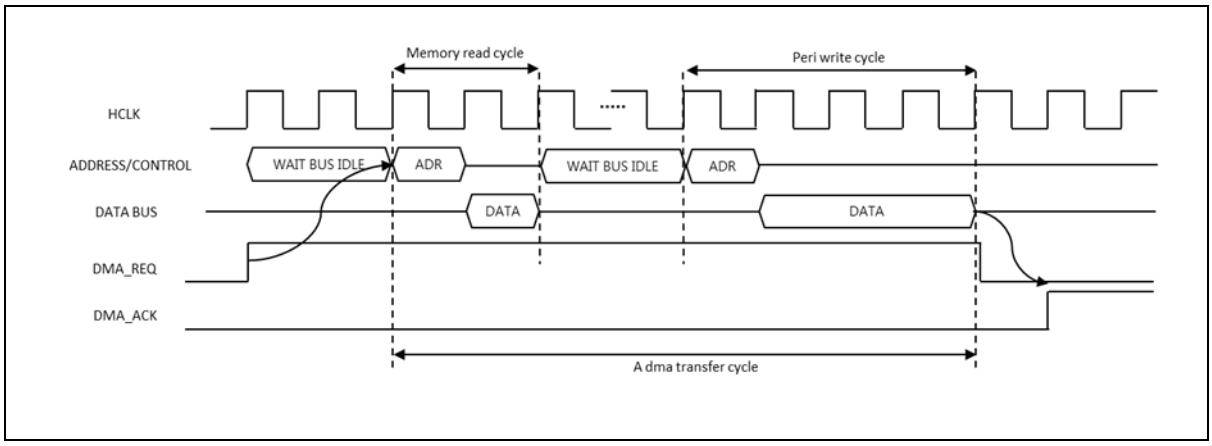


Figure 43. Timing Diagram of DMAC Transfer from Memory to Peripheral

Figure 44 introduces an example of N data transfers with DMA. The DMA transfer is started when DCnSR.DMAEN is set. When all the number of transfer is completed, the DCnSR.DMAEN register will be cleared.

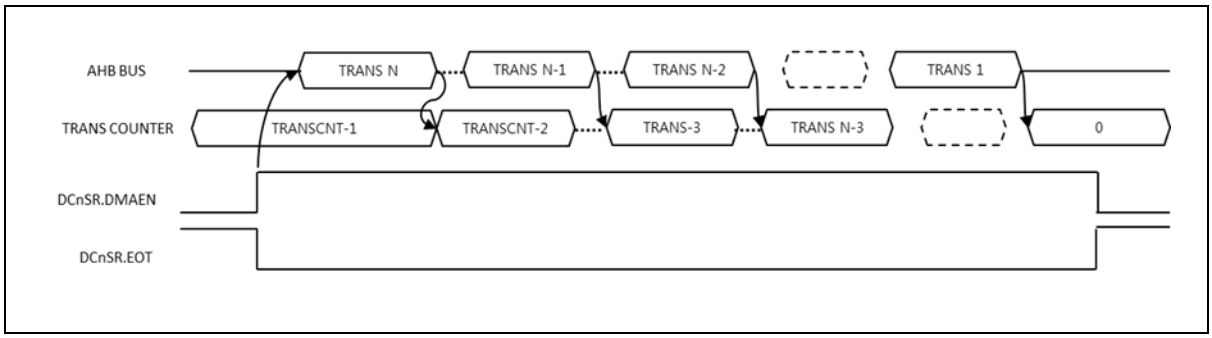


Figure 44. Timing Diagram Example of N DMAC Transfer

8. Static memory controller (EBI)

Static memory controller is used for external bus interfaces to control external memory I/Os. External memory area is divided to 4 sections. 1MByte memory area is allocated for each section respectively. Memory devices (SRAM, ROM and Flash) or I/O devices can be mapped to the memory section. Bus configurations such as bus width of external device and the number of wait are configurable in each section; a user can attach many kinds of memories or I/O devices.

Table 34 introduces pins assigned for static memory controller.

Table 34. Pin Assignment of Static Memory Controller

Pin name	Type	Description
nCS[3:0]	O	Select memory region from 0 to 3.
nRD	O	Perform read operation from external memory.
nUWR	O	Upper byte write signal. When 16-bit bank is organized using 8bit memory, this pin indicates that D[15:8] is being written to external memory
nLWR/nWE	O	Lower byte write signal. It is used in various ways according to the memory and bank organization. When 8-bit or 16-bit bank is organized using 8-bit memory, this pin indicates that lower byte, D[7:0] is being written to external memory.
nUDS	O	Upper Data Select signal. It is used in various ways according to the memory and bank organization. When 16-bit bank is organized using 16-bit memory, this pin indicates that upper byte is being selected.
nLDS	O	Lower Data Select signal. It is used in various ways according to the memory and bank organization. When 16-bit bank is organized using 16-bit memory, this pin indicates that lower byte is being selected.
nWAIT	I	This pin is an external wait signal pin. If value of this pin becomes "L", wait is added as much as nWAIT maintains "L".
ALE	O	Address Latch Enable In the case of muxed Address/Data access, external address buffer for A[15:0] must be used. Address buffer latch AD[15:0] signals when ALE is high.
A[19:16]	O	Upper Address bus for external memory devices
AD[15:0]	I/O	Address or Data input/output bus for external memory devices

8.1 Block diagram

In this section, static memory controller is described in Figure 45.

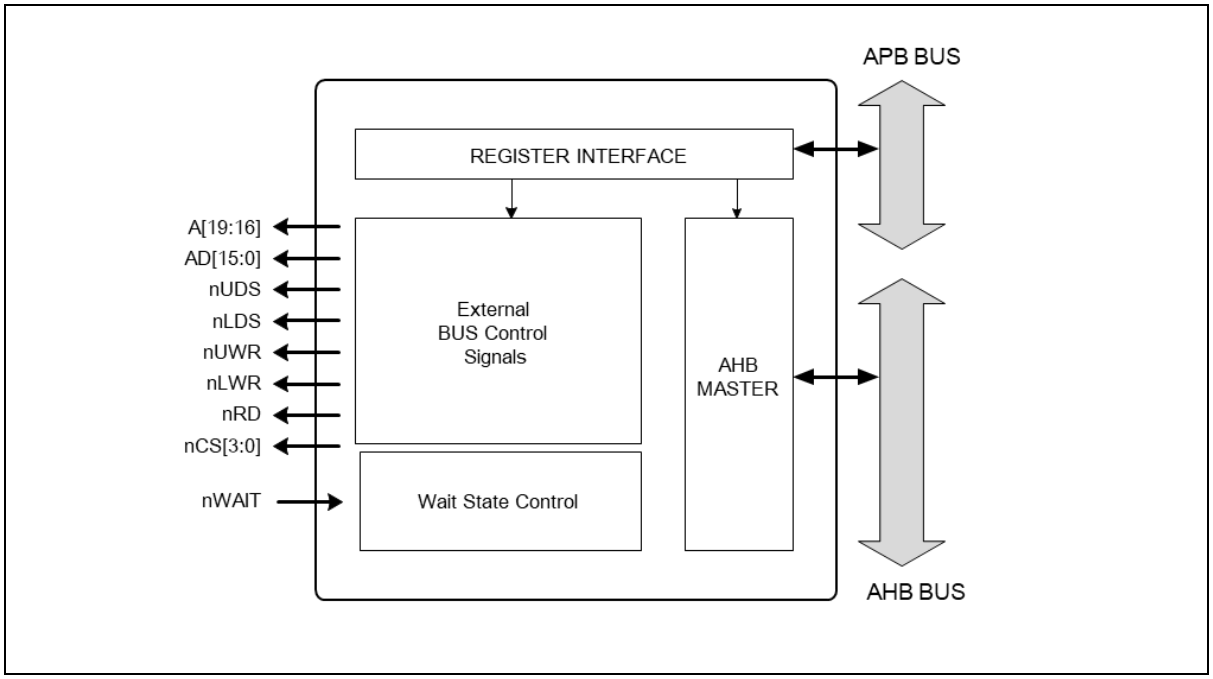


Figure 45. Static Memory Controller Block Diagram

8.2 Registers

Base address of static memory controller is introduced in the followings:

Table 35. Base Address of Static Memory Controller

Name	Base address
Static memory controller	0x4000_6200

Table 36. SMIF Register Map

Name	Offset	Type	Description	Reset value	Reference
SMIBCR0	0x0000	RW	nCS[0] memory area config. register	0x00011F3F	8.2.1
SMIBCR1	0x0004	RW	nCS[1] memory area config. register	0x00011F3F	
SMIBCR2	0x0008	RW	nCS[2] memory area config. register	0x00001F3F	
SMIBCR3	0x000C	RW	nCS[3] memory area config. register	0x00001F3F	

8.2.1 SMIBCRn: Memory area configuration register n

BCRn registers are memory area configuration register. Register size is 32-bit.

SMIBCR0=0x4000_6200, SMIBCR1=0x4000_6204
SMIBCR2=0x4000_6208, SMIBCR3=0x4000_620C

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
IDLE	Reserved	PREIDL	Reserved	EWEN	EWP	IOE	BLEN	AH	MWIDTH	Reserved	NORMWAIT
00	-	00	-	0	0	0	0	0	0	-	0x3F
RW	-	RW	-	RW	RW	RW	RW	RW	RW	-	RW

31 30	IDLE	Insert IDLE cycle at memory access time	
		00	No additional cycle
		01	Insert 1 IDLE cycle
		10	Insert 2 IDLE cycle
		11	Insert 3 IDLE cycle
27 26	PREIDL	Changing falling timing of RD and nUWR/nLWR	
		00	No additional cycle
		01	Extend 1 clock cycle
		10	Extend 2 clock cycle
		11	Extend 3 clock cycle
23	EWEN	0	Ignore external input nWAIT
		1	Enable external input nWAIT request
22	EWP	0	nWAIT polarity is active low
		1	nWAIT polarity is active high
21	IOE	0	Separated Address/Data access
		1	Enable muxed Address/Data access
20	BLEN	0	Disable byte lane selection in case of 8bit memory organization
		1	Enable byte lane selection in case of 8bit memory organization
19 18	AH	When accessing A/D mux I/O area, "AH" represents the time (the number of clock cycle) while the address is valid. Address keeps the value for "AH + 1" clock (HCLK) cycle.	
		00	Keep the address for 1 clock cycle
		01	Keep the address for 2 clock cycle
		10	Keep the address for 3 clock cycle
		11	Keep the address for 4 clock cycle
17 16	MWIDTH	00	External memory bus is 8bit width
		01	External memory bus is 16bit width
5 0	NORMWAIT	It represents wait value to be used for memory read/write operation. 1-64 wait is supported and wait value is NORMWAIT+1.	

NOTES:

1. Memory read/write access time. 1-64 wait is supported and wait value NORMWAIT – PREIDL.
2. NORMWAIT must be greater than PREIDL (NORMWAIT > PREIDL).

8.3 Functional description

External memory register is organized with 4 sections. Each section consists of 1MB address region. A[26:24] is used to select memory region and A[19:0] is used in each memory region.

Table 37 provides memory map information regarding internal address. External memory region offset is 0x60000000 in the memory map.

Table 37. Memory Region Selection and nCS Output by Address

A[26:24]	nCS[3:0]	Memory region
000	1110	0x6000_0000 to 0x600F_FFFF
001	1101	0x6100_0000 to 0x610F_FFFF
010	1011	0x6200_0000 to 0x620F_FFFF
011	0111	0x6300_0000 to 0x630F_FFFF

8.3.1 Chip select signal

Chip select signal, nCS[3:0] becomes "L" when corresponding memory region is selected, and it has a various signal width according to the IDLE field of the SMIBCRn. Chips select signal is changed at the rising edge of internal bus clock.

8.3.2 Byte lane control and control signal

Data access size of CPU or internal bus master can be 8-bit, 16-bit or 32-bit. SMI controller should sort the data form external bus. When bus master accesses external memory region, SMI determines which byte will be used between upper and lower byte according to bus control register.

Table 38 shows data is sorted by memory access size. Lower byte D[7:0] is used for 8-bit access, and upper byte D[15:8] is used for 16-bit access. Figure 46 shows that sort in 8-bit memory bus system. In this case, only D[7:0] is used regardless of memory access size.

Table 38. Data Sort for 8-bit Memory Bus

Internal access width	External access cycle	Upper byte lane D[15:8]	Lower byte lane D[7:0]
Byte size access	1st EBI BUS Cycle	Not valid	Valid for byte A[1:0]
Half word size access	1st EBI BUS Cycle	Not valid	Valid for byte A[0] = 0
	2nd EBI BUS Cycle	Not valid	Valid for byte A[0] = 1
Word size access	1st EBI BUS Cycle	Not valid	Valid for byte A[1:0] = 00
	2nd EBI BUS Cycle	Not valid	Valid for byte A[1:0] = 01
	3rd EBI BUS Cycle	Not valid	Valid for byte A[1:0] = 10
	4th EBI BUS Cycle	Not valid	Valid for byte A[1:0] = 11

Table 39 shows 16-bit memory bus system. In the case of byte access, it is determined which byte will be used between D[7:0] and D[15:8]. In case of half word or word access, D[15:0] and D[7:0] take upper byte and lower byte.

Table 39. Data Sort for 16-bit Memory Bus

Internal access width	External access cycle	Upper byte lane D[15:8]	Lower byte lane D[7:0]
Byte size access	1st EBI BUS Cycle	Not valid	Valid for byte A[0] = 0
	1st EBI BUS Cycle	Valid for byte A[0] = 1	Not valid
Half word size access	1st EBI BUS Cycle	Valid for upper byte	Valid for lower byte
Word size access	1st EBI BUS Cycle	Valid for upper byte A[1] = 0	Valid for lower byte A[1] = 0
	2nd EBI BUS Cycle	Valid for upper byte A[1] = 1	Valid for lower byte A[1] = 1

Signals nUWR and nLWR are generated according to the bus width, type of memory and A[0]. Table 40 shows example accessing 8-bit memory using 16-bit memory bus D[15:0].

Table 40. Config Signal by A[0]

Internal access size	A[0]	nUWR	nLWR	Access size
Half word	X	L	L	16-bit
Byte	0	H	L	8-bit
Byte	1	L	H	8-bit

Figure 46 shows bus configuration with 8-bit memory type. Signals nUWR and nLWR are used for 8-bit or 16-bit bus with 8-bit memory type.

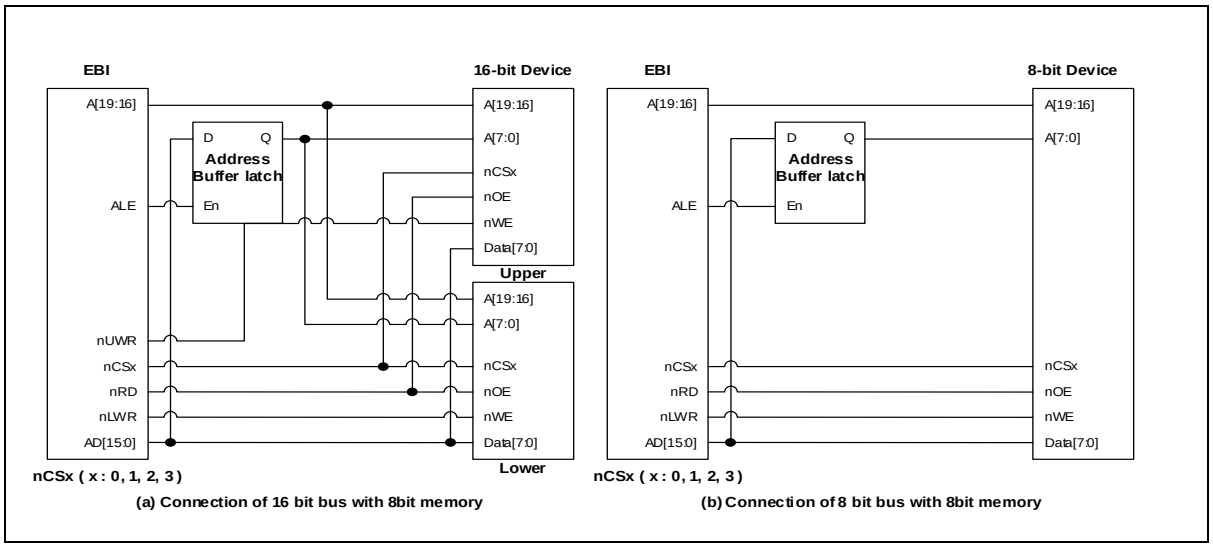
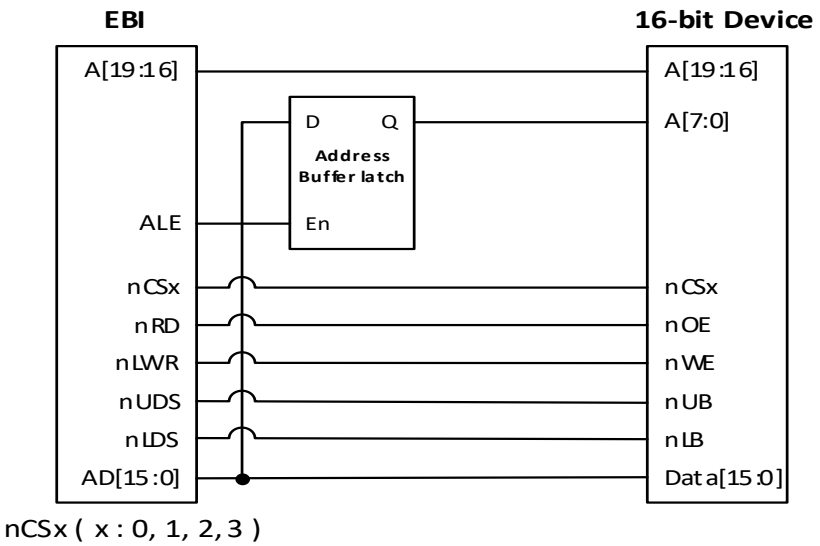


Figure 46. Bus Configuration with 8-bit Memory

Figure 47 describes memory configuration with 16-bit memory. In this case, byte selection signals, nUDS and nLDS are required.



(a) Connection of 16 bit bus with 16bit memory

Figure 47. Bus Configuration with 16-bit Memory

As seen in Figure 47, with 8-bit memory, regardless of the bus width, nLWR and nUWR are connected for write operation, and nRD is connected for read operation. With 16-bit memory, nWE and nRD are connected in write and read operation. nUDS and nLDS are connected in both read and write operations. nUDS and nLDS are byte selection signals of memory.

8.3.3 Bus control signal in read operation

Read operation by SMI is initiated after wait cycle of NORMWAIT in SMIBCRn register.

Figure 48 shows word read operation in case of "NORMWAIT =1". Word read is possible with 4 times of byte access because 8-bit bus is used.

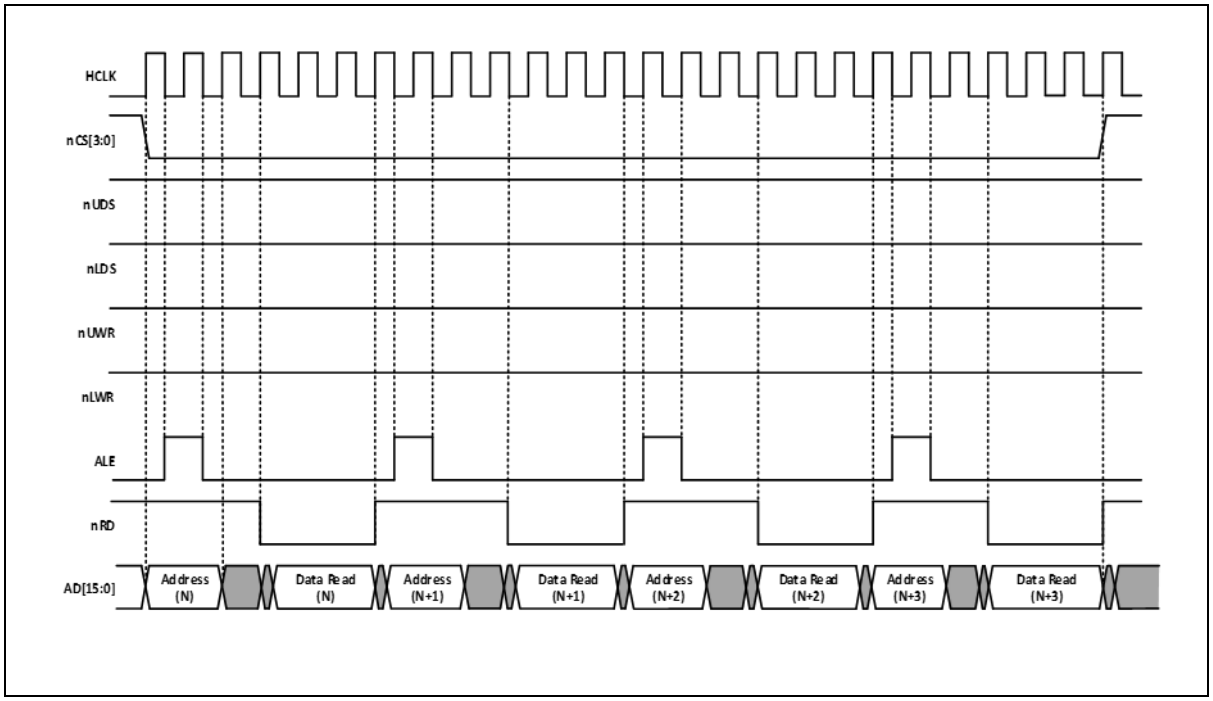


Figure 48. Word Read Operation with 8-bit Memory and 8-bit Bus

Figure 49 shows half word read operation when "NORMWAIT = 2". Like the previous case, half word memory read is possible with 2 times of byte memory access because only 8-bit bus is used.

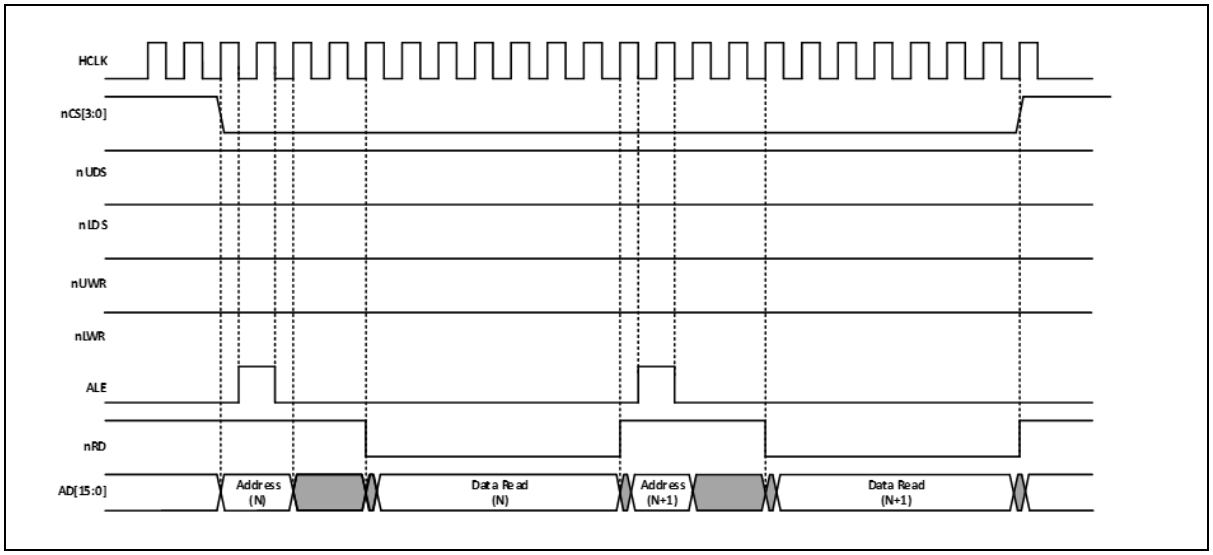


Figure 49. Half Word Read Operation with 8-bit Memory and 8-bit Bus

Figure 50 shows word read operation when "NORMWAIT = 2". D[15:0] is used as data bus because 16-bit bus is used. Word read is possible with 2 times of memory access.

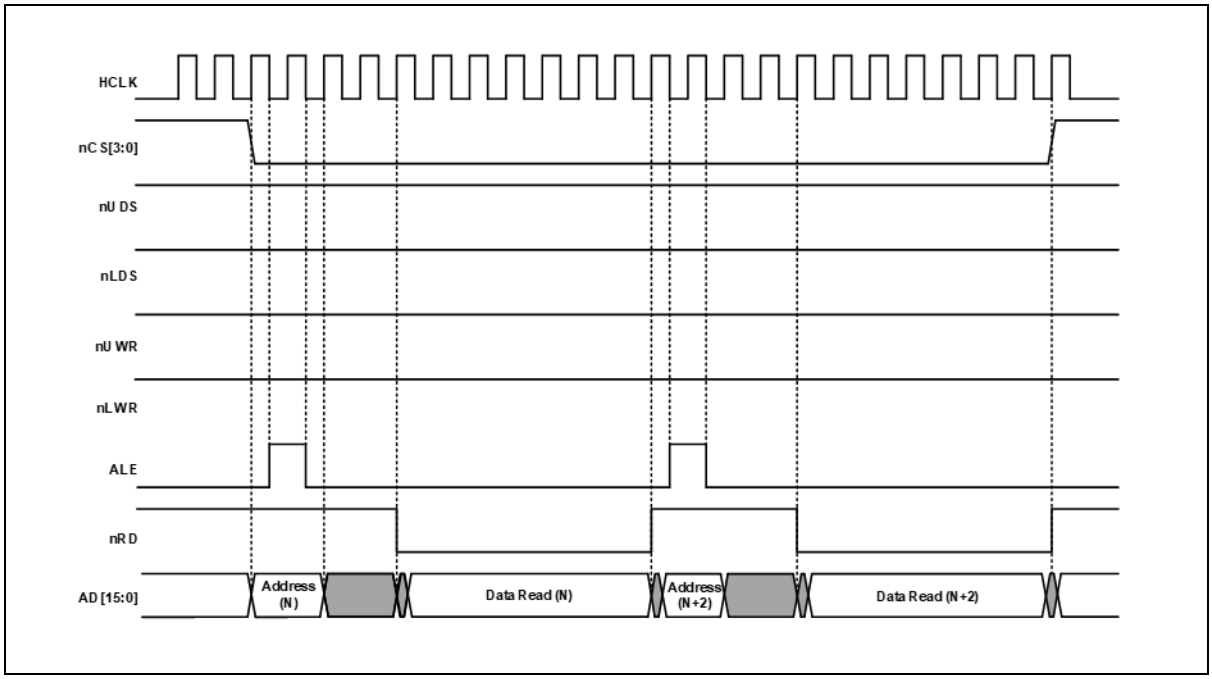


Figure 50. Word Read Operation with 8-bit Memory and 16-bit Bus

Figure 51 shows half word read operation when "NORMWAIT = 2". D[15:0] is used as data bus because 16-bit bus is used. Half word read is possible with 1 times of memory access.

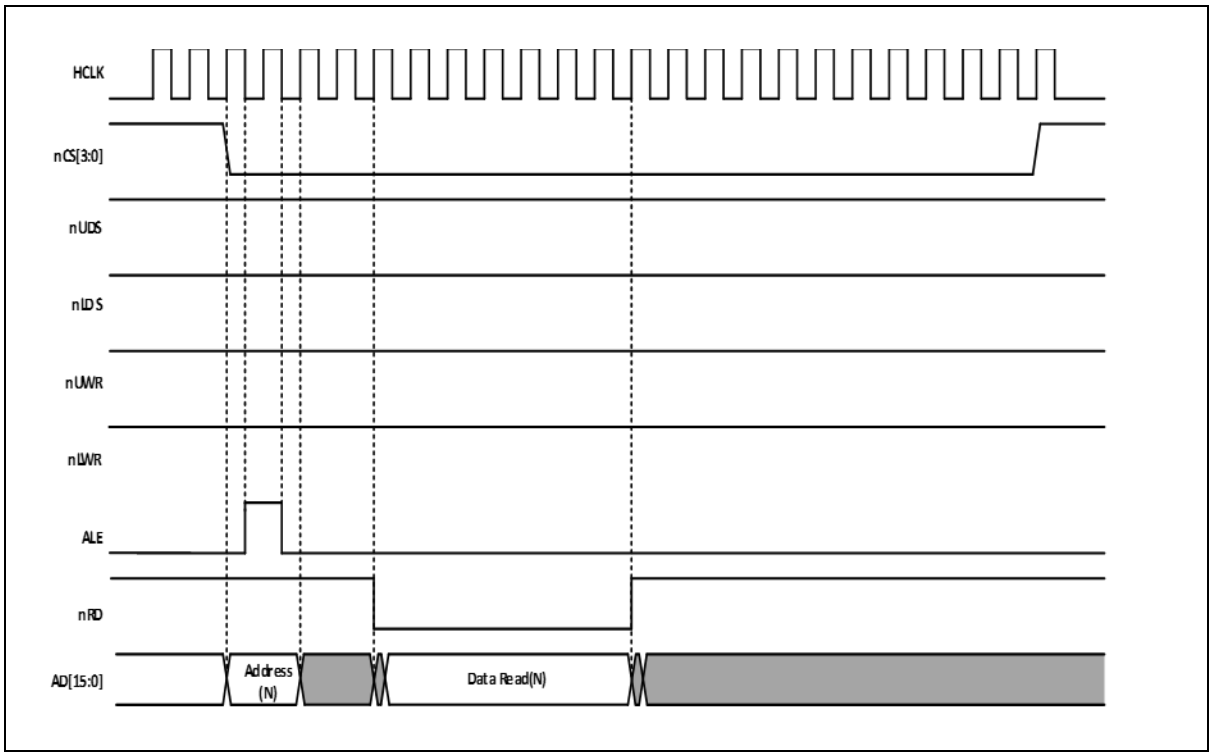


Figure 51. Half Word Read Operation with 8-bit Memory and 16-bit Bus

Figure 52 and Figure 53 show word and half word read operation when "NORMWAIT = 2". D[15:0] is used as data bus for 16-bit bus. Word and half word read are possible with 2 and 2 times of memory access. Because bus is organized with 16-bit memory, upper byte selection signal "nUDS" and lower byte signal "nLDS" is used with nRD signal.

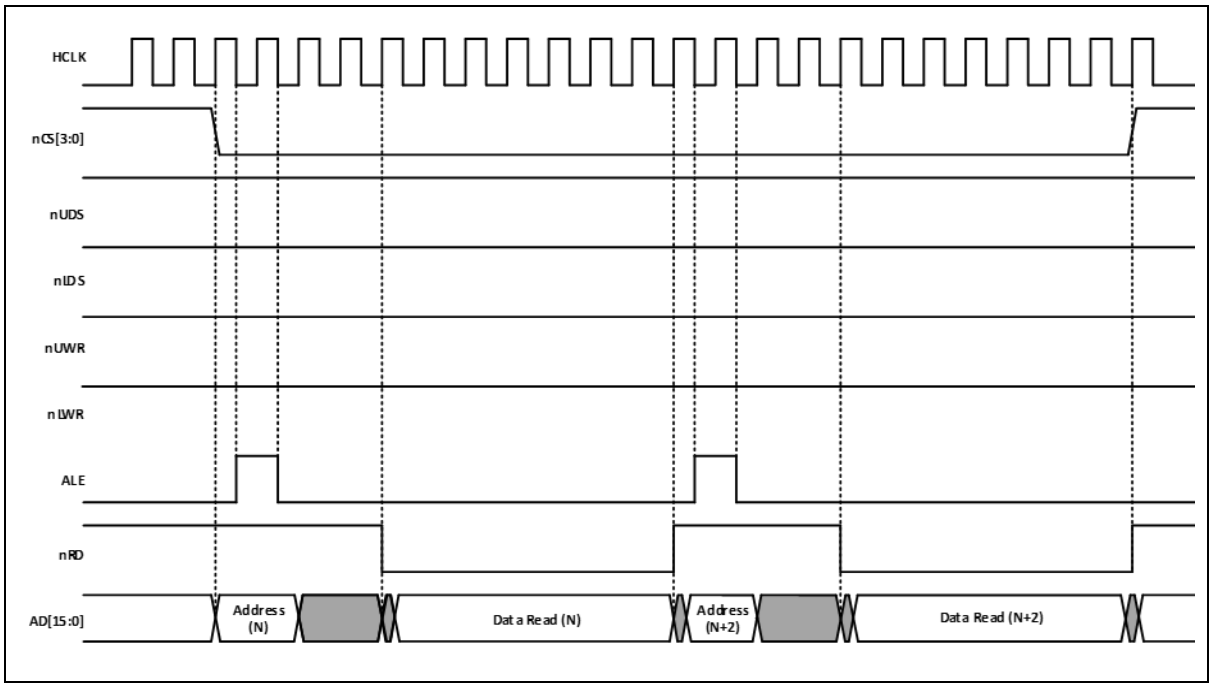


Figure 52. Word Read Operation with 16-bit Memory and 16-bit Bus

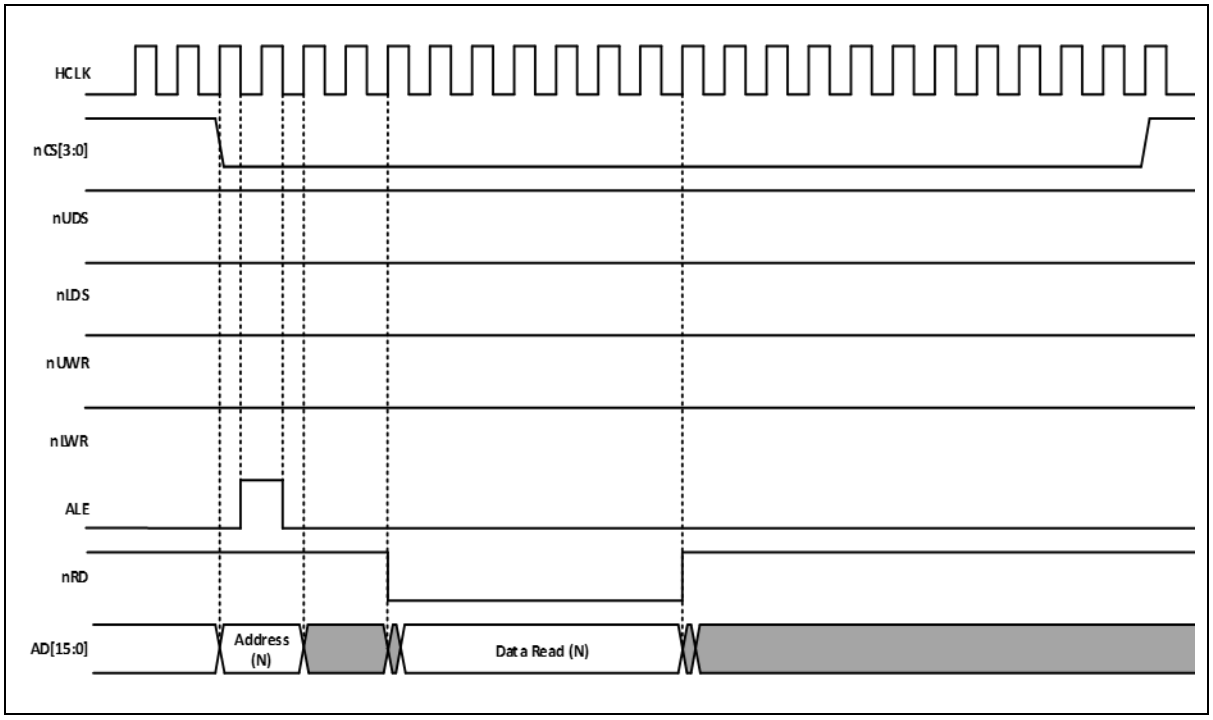


Figure 53. Half Word Read Operation with 16-bit Memory and 16-bit Bus

8.3.4 Bus control signal in write operation

Figure 54 and Figure 55 show word and half word operations when "NORMWAIT = 1" and "NORMWAIT = 2". D[7:0] is used for 8-bit bus. Word and half word write operations are possible through 4 and 2 times of memory access. nUWR will not change to "L" because 8-bit bus is used.

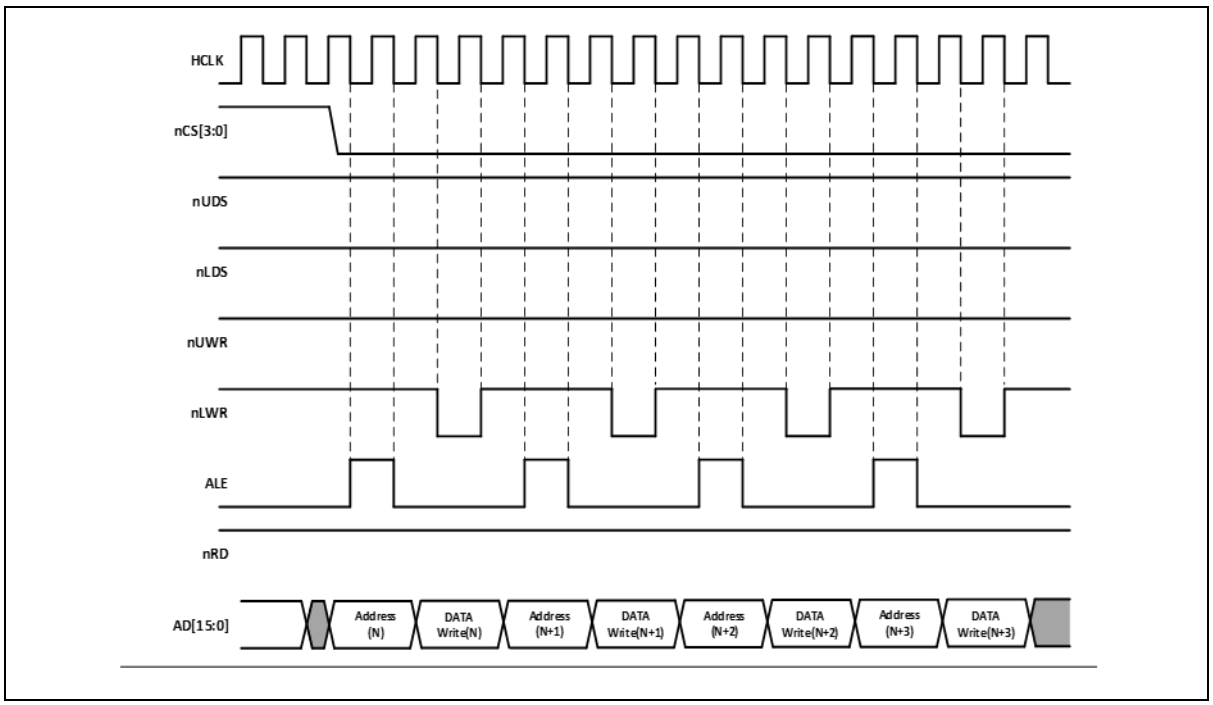


Figure 54. Word Write Operation with 8-bit Memory and 8-bit Bus

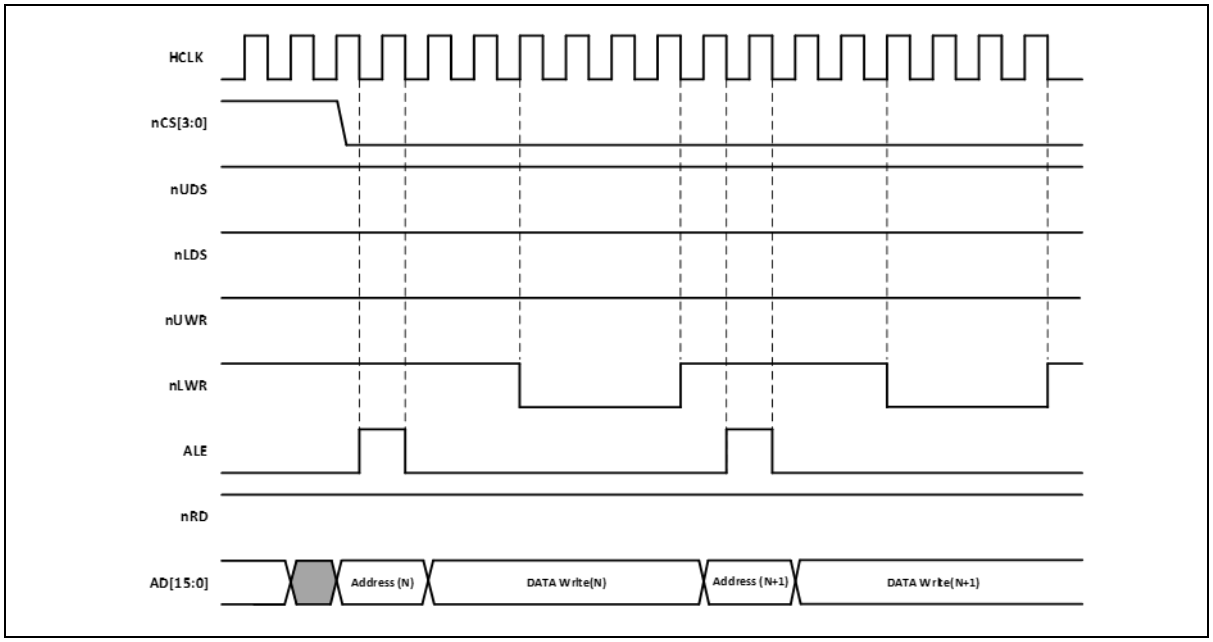


Figure 55. Half Word Write Operation with 8-bit Memory and 8-bit Bus

Figure 56 and Figure 57 show word and half word write operations when "NORMWAIT= 2". D[7:0] is used for 16-bit bus. Word and half word writes are possible with 2 and 1 times of memory access.

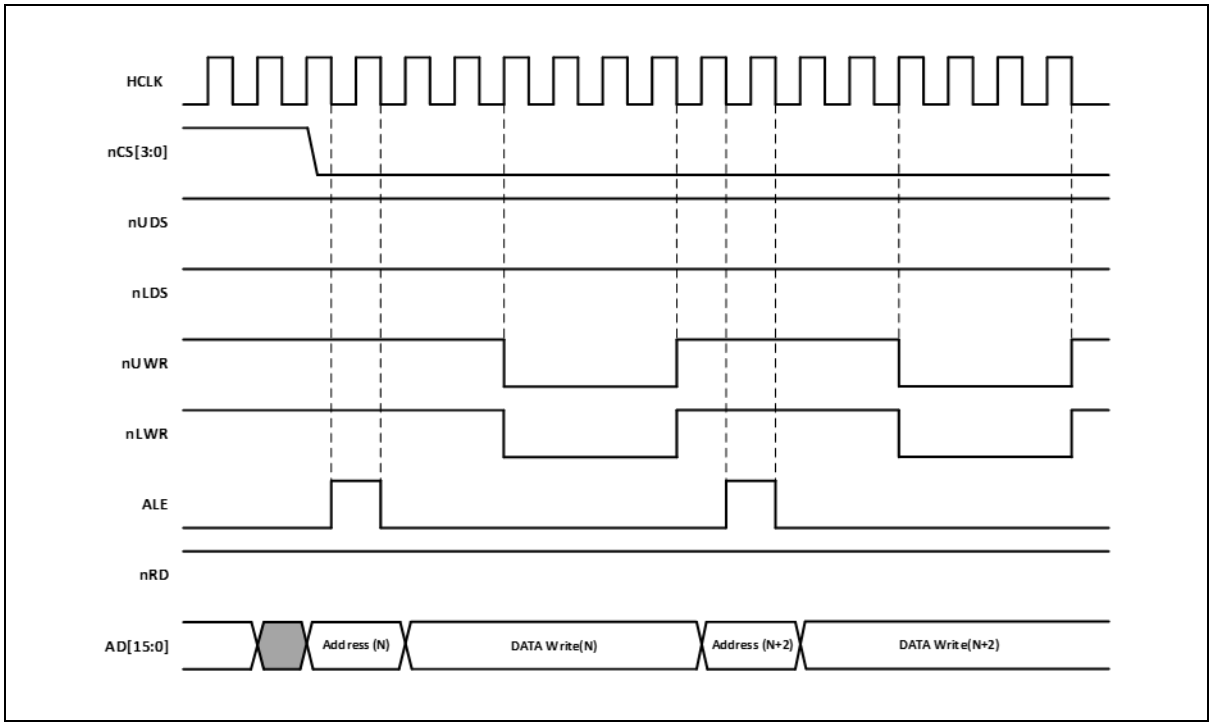


Figure 56. Word Write Operation with 8-bit Memory and 16-bit Bus

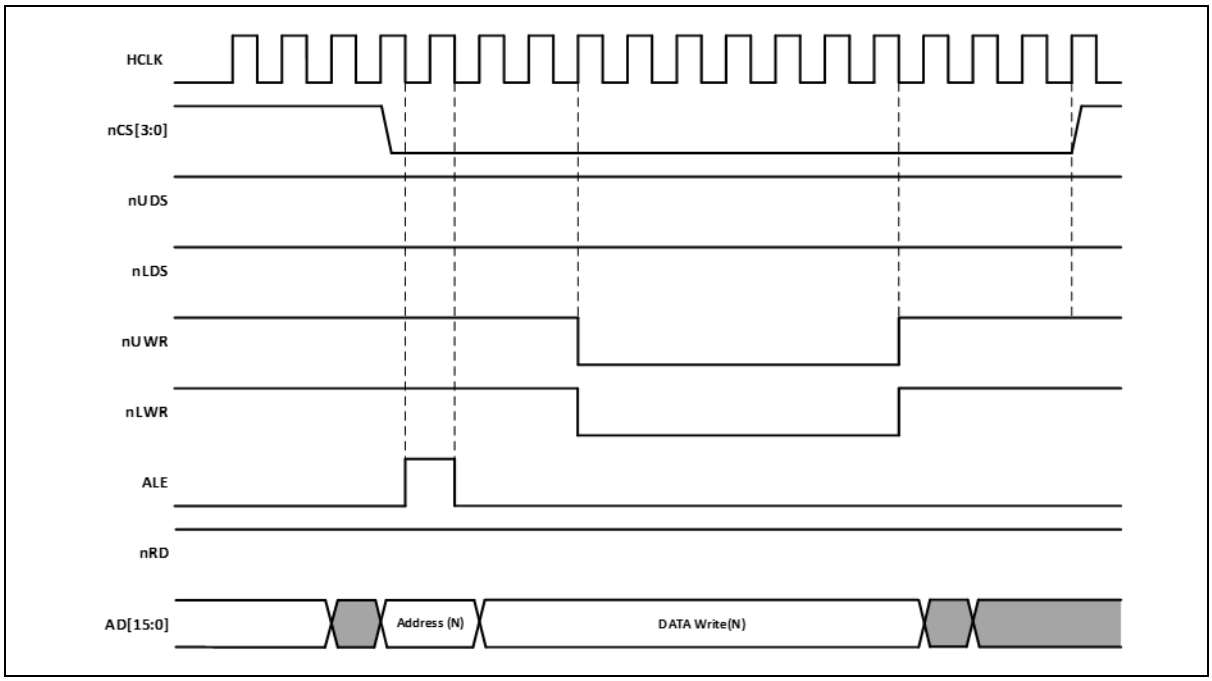


Figure 57. Half Word Write Operation with 8-bit Memory and 16-bit Bus

Figure 58 and Figure 59 show word (half word) write operations when "NORMWAIT = 2". D[15:0] is used for 16-bit bus. Word (half word) write is possible with 2(1) times of memory access. Byte select signals (nUDS, nLDS) are used with nRD and nWE signals because 16-bit memory is used.

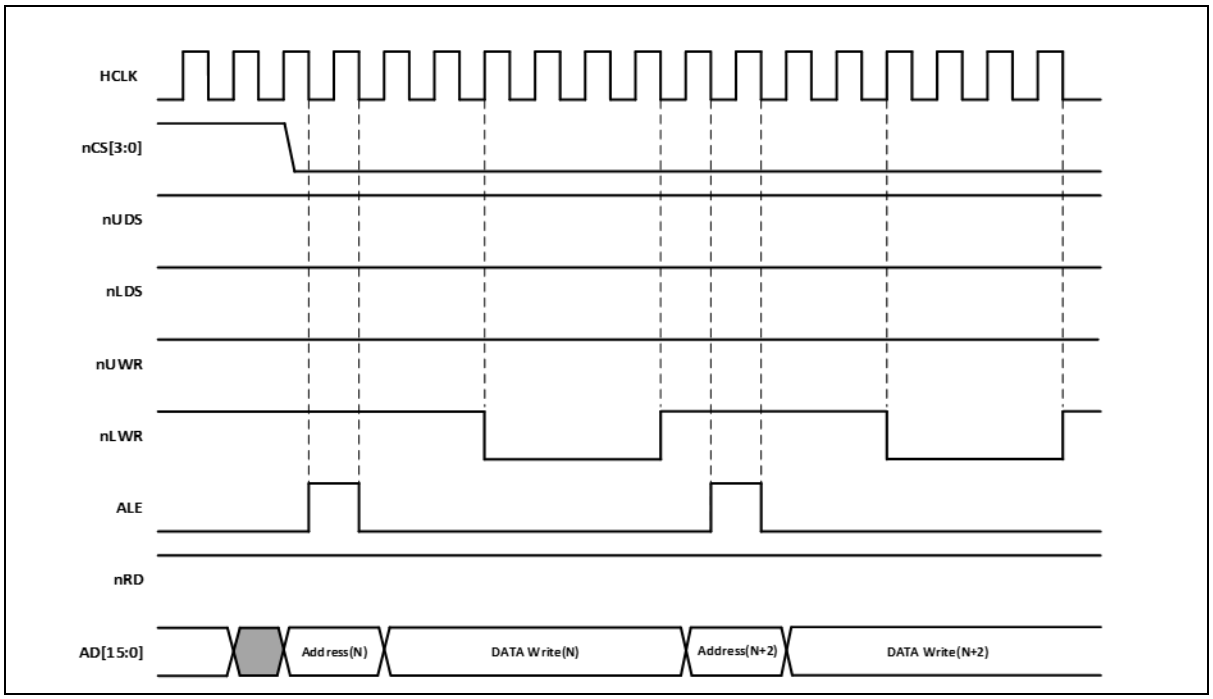


Figure 58. Word Write Operation with 16-bit Memory and 16-bit Bus

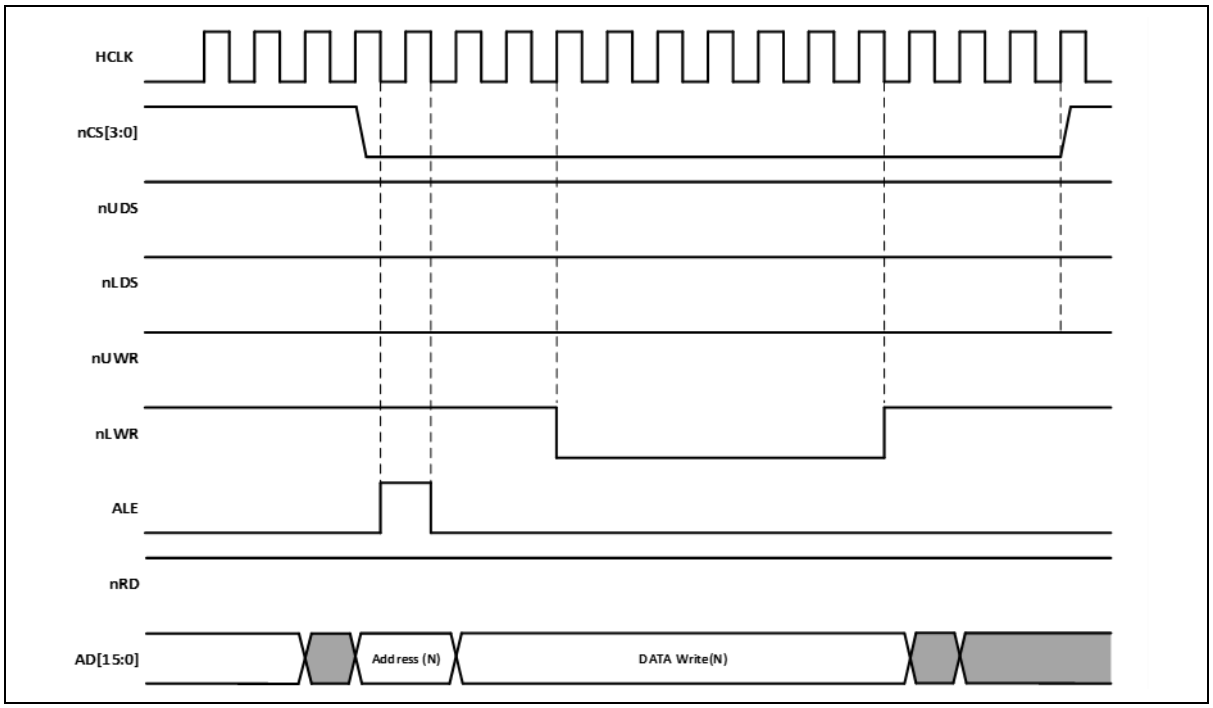


Figure 59. Half Word Write Operation with 16-bit Memory and 16-bit Bus

8.3.5 Timing change of control signals

As setting IDLE field, it is possible to change IDLE cycle. This equals to keeping "H" as long as cycle defined by changing nCS from "L" to "H" as shown in Figure 60 and Figure 61. nRD/nUWR/nLWR signals are possible to be extended up to 3 clock cycles by setting PREIDL field.

Figure 60 shows that nCS keeps "H" after memory access when IDLE field is set to "1". This represents that additional cycle is inserted as much as the number of IDLE.

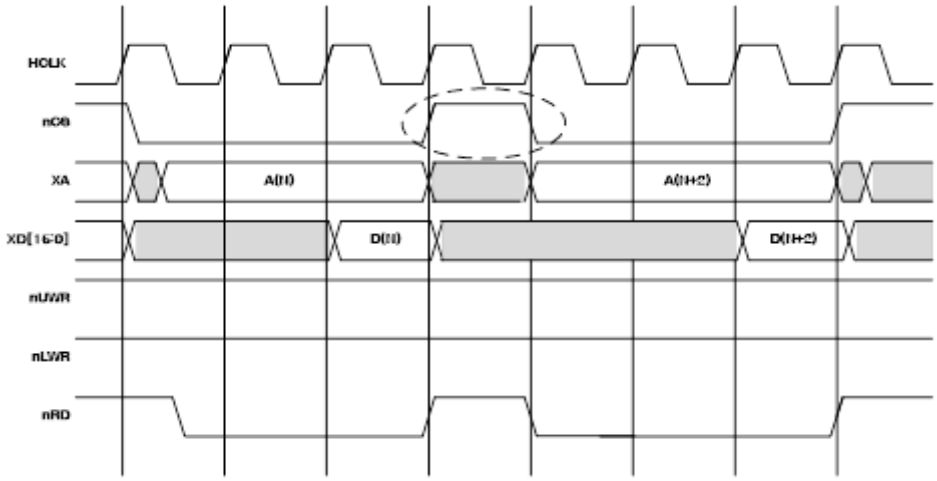


Figure 60. Read Timing Changing by 'IDLE = 1'

Figure 61 shows that nLWR and nUWR are delayed as much as the number of PREAMBLE when PREIDL is "2". In case of read operation, nRD is delayed as much as PREAMBLE.

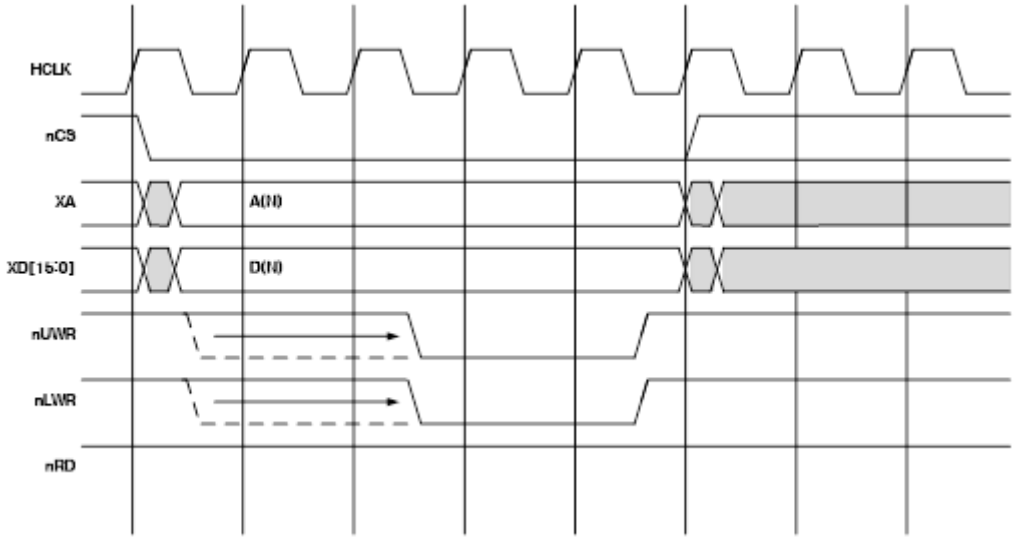


Figure 61. Write Timing Changing by 'PREIDL = 1'

8.3.6 Wait control while accessing memory

Internal wait is applied as much as NORMWAIT value while accessing memory and bus cycle can be extended by adding external wait using nWAIT input. Internal wait ranges from 1 to 64. It is applied automatically whenever accessing memory.

nWAIT becomes active according to EWP (External nWAIT Polarity). The bus enters in wait state internally until nWAIT becomes inactive. If EWO is "0", external wait is active when nWAIT is "L". If EWP is "1", external wait is active when nWAIT is "H".

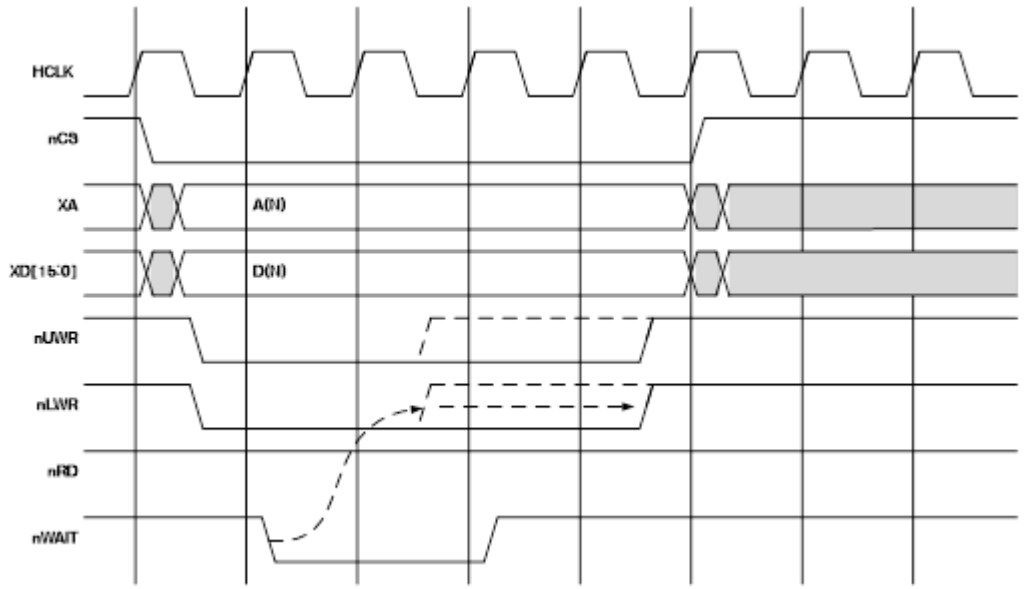


Figure 62. Write Operation with External Wait (nWAIT), 'EXP = 0'

9. Watchdog timer (WDT)

Watchdog timer (WDT) rapidly detects CPU's malfunction such as endless looping caused by noise, and resumes the CPU to the normal state. WDT signal for malfunction detection can be used as either a CPU reset or an interrupt request. When WDT is not being used for malfunction detection, it can be used as a timer generating an interrupt at fixed intervals.

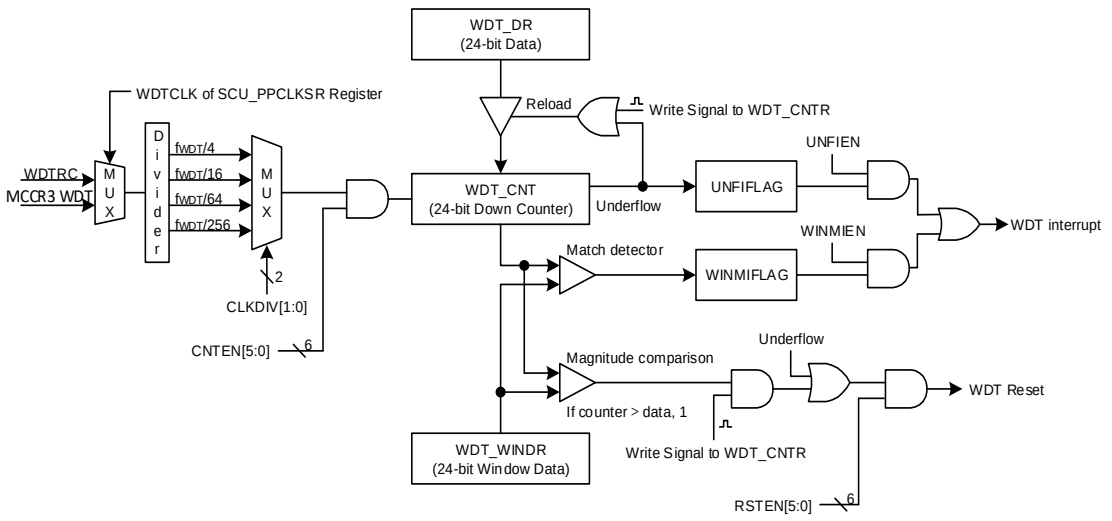
When WDT_CNT value is reached to WDT_WINDR value, a watchdog interrupt can be generated. The underflow time of WDT can be set by WDT_DR. If the underflow occurs, an internal reset is generated. WDT operates on the WDTRC embedded RC oscillator clock.

WDT of A31G32x series features followings:

- 24-bit down counter (WDT_CNT)
- Select reset or periodic interrupt
- Count clock selection
- Watchdog overflow output signal
- Counter window function

9.1 WDT block diagram

In this section, WDT block diagram is introduced in Figure 63.



NOTE:

1. If a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT register after 1 clock delay with the clock which is selected by the CLKDIV bits.

Figure 63. WDT Block Diagram

9.2 Registers

Base address of WDT is introduced in the followings:

Table 41. Base Address of WDT

Name	Base address
WDT	0x4000_1A00

Table 42. WDT Register Map

Name	Offset	Type	Description	Reset value	Reference
WDT_CR	0x0000	RW	Watch-dog Timer Control Register	0x0000_0000	9.2.1
WDT_SR	0x0004	RW	Watch-dog Timer Status Register	0x0000_0080	9.2.2
WDT_DR	0x0008	RW	Watch-dog Timer Data Register	0x0000_3D09	9.2.3
WDT_CNT	0x000C	RO	Watch-dog Timer Counter Register	0x0000_0FFF	9.2.4
WDT_WINDR	0x0010	RW	Watch-dog Timer Window Data Register	0x0000_FFFF	9.2.5
WDT_CNTR	0x0014	WO	Watch-dog Timer Counter Reload Register	0x0000_0000	9.2.6

9.2.1 WDT_CR: Watchdog timer control register

WDT module should be configured properly before running. WDT module can make reset event or assert interrupt signal to system. Size of this register is 32-bit.

WDT_CR=0x4000_1A00

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
WTIDKY	RSTEN	CNTEN	WINMIEN	UNFIEN	CLKDIV
0x0000	000000	000000	0	0	00
WO	RW	RW	RW	RW	RW

31 16	WTIDKY	Write Identification Key.	
		On writes, write 0x5A69 to these bits, otherwise the write is ignored.	
15 10	RSTEN	Watch-dog Timer Reset Enable bits.	
		0x25	Disable watch-dog timer reset.
		Others	Enable watch-dog timer reset.
9 4	CNTEN	Watch-dog Timer Counter Enable bits.	
		0x1A	Disable watch-dog timer counter.
		Others	Enable window data match interrupt.
3	WINMIEN	Watch-dog Timer Window Match Interrupt Enable bit.	
		0	Disable window data match interrupt.
		1	Enable window data match interrupt.
2	UNFIEN	Watch-dog Timer Underflow Interrupt Enable bit.	
		0	Disable watch-dog timer underflow interrupt.
		1	Enable watch-dog timer underflow interrupt.
1 0	CLKDIV	Watch-dog Timer Clock Divider bits, The clock which is selected by SCU_PPCLKSR[0].	
		00	fWDT/4
		01	fWDT/16
		10	fWDT/64
		11	fWDT/256

9.2.2 WDT_SR: Watchdog timer status register

WDT_SR register is 32-bit size, and able to do 32/16/8-bit access.

WDT_SR=0x4000_1A04

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			DBGCNTEN	Reserved	WINMIFLAG	UNFIFLAG
-			1	-	0	0
-			RW	-	RW	RW

7	DBGCNTEN	Watch-dog Timer Counter Enable bit When the core is halted in the debug mode.	
		0	The watch-dog timer counter continues even if the core is halted.
		1	The watch-dog timer counter is stopped when the core is halted.
		Note: This bit is set to "1b" by POR reset.	
1	WINMIFLAG	Watch-dog Timer Window Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
0	UNFIFLAG	Watch-dog Timer Underflow Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1.

NOTES:

1. Window match flag of WDT
2. Window match flag is recommended for System Clock(fx) ≥ 2*WDT Clock(WDT)

9.2.3 WDT_DR: Watchdog timer data register

WDT_DR register is used to update WDT_CNT register. Size of this register is 32-bit.

WDT_DR=0x4000_1A08

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	DATA		
-	0x003D09		
-	RW		

23 0	DATA	Watch-dog Timer Data bits. The range is 0x000000 to 0xFFFFFF.

NOTE:

1. Once any value is written to this data register, the register can't be changed until a system reset.

9.2.4 WDT_CNT: Watchdog timer counter register

WDT_CNT register represents the current count value of 32-bit down counter. When the counter value reach to 0, the interrupt or reset will be asserted. Size of this register is 32-bit.

WDT_CNT=0x4000_1A0C

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	CNT		
-	0x000FFF		
-	RO		

23 0	CNT	Watch-dog Timer Counter bits.

9.2.5 WDT_WINDR: Watchdog timer window data register

WDT_WINDR register is used to compare with WDT_CNT for WINDOW function. Size of this register is 32-bit.

WDT_WINDR=0x4000_1A10

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	WDATA		
-	0x00FFFF		
-	RW		

23 0	WDATA	Watch-dog Timer Data bits. The range is 0x000000 to 0xFFFFFF.

NOTE:

1. Once any value is written to this data register, the register can't be changed until a system reset.

9.2.6 WDT_CNTR: Watchdog timer counter reload register

WDT_CNTR register is used to make reload signal. If reload signal is I, WDT_DR Value is reloaded to WDT_CNT. Size of this register is 32-bit.

WDT_CNTR=0x4000_1A14

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			CNTR
-			0x00
-			WO

7 0	CNTR	Watch-dog Timer Counter Reload bits.	
		0x6A	Reload the WDTDR value to watch-dog timer counter and re-start. (Automatically cleared to "0x00" after operation)
		Others	No effect

9.3 Functional description

WDT counter can be enabled by CNTEN (WDT_CR[9:4]). Corresponding bit field of the register CNTEN is set with any value other than 0x1A. As WDT activates, a down counter will start counting from loaded value.

If RSTEN (WDT_CR[15:10]) is set with any value other than 0x25, WDT reset would be asserted when the WDT counter value reaches to '0' (underflow event) from WDT_DR value. Before WDT counter reaches to '0', software can write 0x6A to WDT_CNTR register in order to reload WDT counter when the counter value is less than or equal to the value of window data register. WDT reset may be asserted if the reload occurs when counter is greater than window data.

9.3.1 Timing diagram

In this section, WDT interrupt and reset timing diagram is introduced in Figure 64.

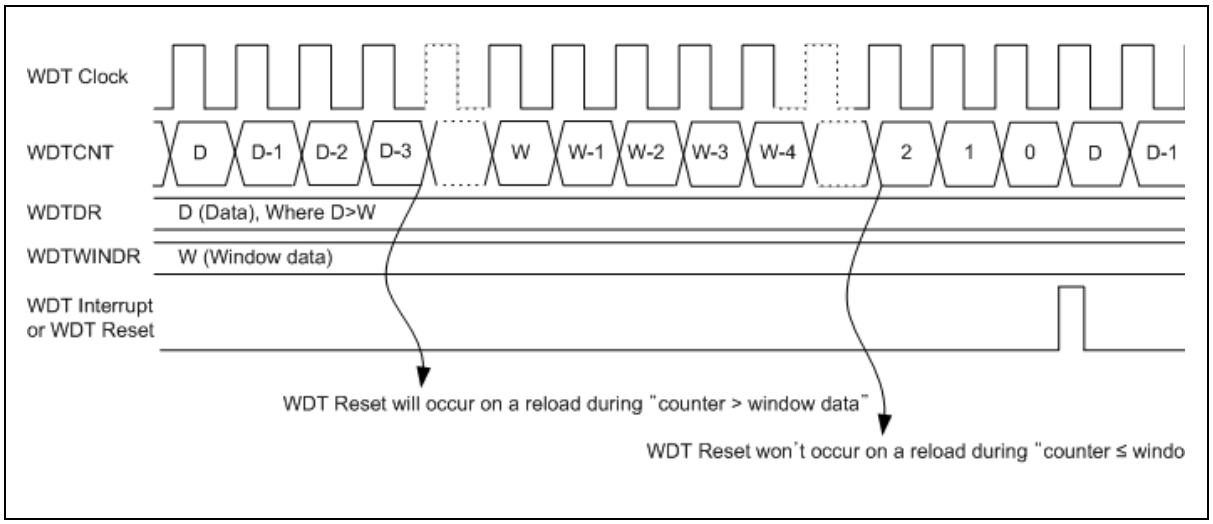


Figure 64. WDT Interrupt and WDT Reset Timing Diagram

9.3.2 Prescale table

WDT includes a 24-bit down counter with programmable pre-scaler to define different time-out intervals.

Clock sources of WDT can be WDT_RC or PCLK. PCLK can be selected by WDTCLK (SCU_PPCLKSR[0]) which is set to '1', then the WDTCNFIG[2] bit of configure option page 1 is cleared to logic "0b".

To configure a WDT counter as a base clock, user can control 2-bit pre-scaler CLKDIV [1:0] in the WDT_CR register, and the maximum pre-scaled value is "clock source frequency/256". The pre-scaled WDT counter clock frequency values are listed in the following Table 43.

Selectable clock source (40KHz to 48MHz) and time-out interval at a single count

Time-out period = (Load Value + 1) * (1/pre-scaled WDT counter clock frequency)

***Time out period (when the Load Value reaches 0, underflow flag is set to '1')**

Table 43. Pre-scaled WDT Counter Clock Frequency

Clock source	WDTCLKIN	WDTCLKIN/4	WDTCLKIN/16	WDTCLKIN/64	WDTCLKIN/256
WDTRC	31.250KHz	7.8125KHz	1.953125KHz	488.28125Hz	122.0703125Hz
MCCR3 WDT	MCCR3 WDT	MCCR3 WDT/4	MCCR3 WDT/16	MCCR3 WDT/64	MCCR3 WDT/256

10. Watch timer

Watch timer (WT) has functions for RTC (Real Time Clock) operation. It is generally used for RTC design. WT consists of a clock source select circuit, a timer counter circuit, an output select circuit and watch timer control registers.

Prior to operate WT, a user needs to determine an input clock source and output interval, and to set WTEN as '1' in watch timer control register (WT_CR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WT_CR register.

Watch timer counter circuit corporates a 26-bit counter. Low 14 bits of the counter form a binary counter and high 12 bits form an auto reload counter in order to raise resolution. In WTCLR, it can control WT clear and set interval value at write time, and it can read 12-bit WT counter value at read time.

10.1 WT block diagram

As shown in Figure 65, WT of A31G32x series have the following blocks:

- 14-bit divider
- 12-bit up-counter
- RTC function

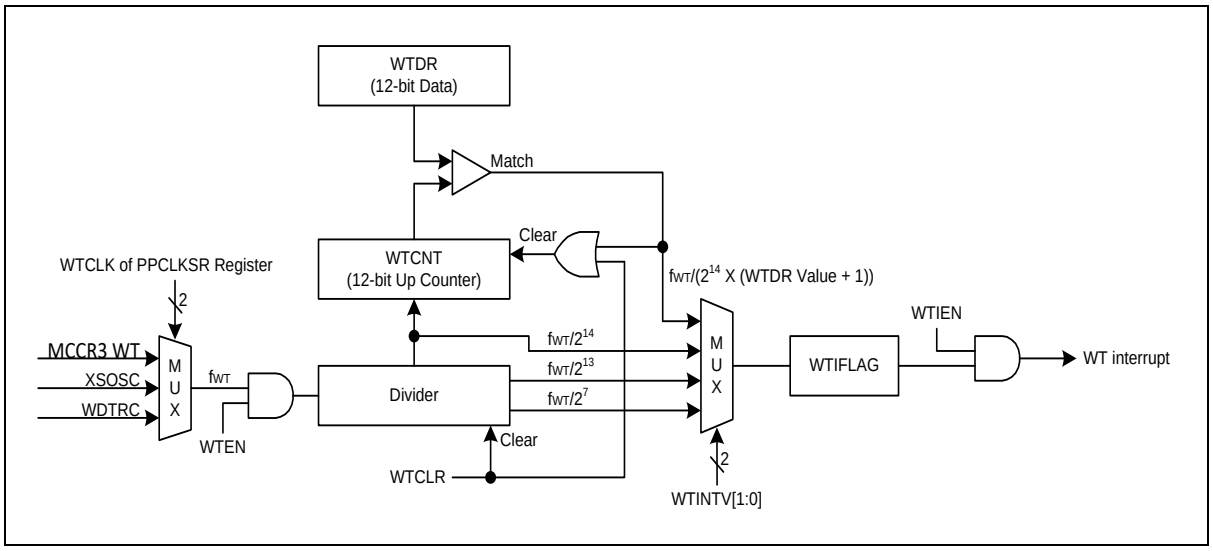


Figure 65. Watch Timer Block Diagram

10.2 Registers

Base address of WT is introduced in the followings:

Table 44. Base Address of WT

Name	Base address
WT	0x4000_2000

Table 45. WT Register Map

Name	Offset	Type	Description	Reset value	Reference
WT_CR	0x0000	RW	Watch Timer Control Register	0x0000_0000	10.2.1
WT_DR	0x0004	RW	Watch Timer Data Register	0x0000_0FFF	10.2.2
WT_CNT	0x0008	RO	Watch Timer Counter Register	0x0000_0000	10.2.3

10.2.1 WT_CR: Watch timer control register

WT_CR is a 32-bit register, and able to do 32/16/8-bit access.

WT_CR=0x4000_2000

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			WTEN	Reserved	WTINTV	WTIEN	Reserved	WTIFLAG	WTCLR
-			0	-	00	0	-	0	0
-			RW	-	RW	RW	-	RW	RW

7	WTEN	Watch Timer Operation Enable bit.	
		0	Disable watch timer operation.
		1	Enable watch timer operation.
5 4	WTINTV	Watch Timer Interval Selection bits.	
		00	$fWT/2^{7}$
		01	$fWT/2^{13}$
		10	$fWT/2^{14}$
		11	$fWT/(2^{14}x(WTDR\ value + 1))$
		NOTE: These bits should be changed during WTEN bit is "0b".	
3	WTIEN	Watch Timer Interrupt Enable bit.	
		0	Disable watch timer interrupt.
		1	Enable watch timer interrupt.
1	WTIFLAG	Watch Timer Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
0	WTCLR	Watch Timer Counter and Divider Clear bit.	
		0	No effect.
		1	Clear the counter and divider (Automatically cleared to "0b" after operation)

10.2.2 WT_DR: Watch timer data register

WT_DR is a 32-bit register.

WT_DR=0x4000_2004

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			WTDATA
-			0xFFF
.			RW

11 0	WTDATA	Watch Timer Data bits. The range is 0x001 to 0xFFF.

10.2.3 WT_CNT: Watch timer counter register

WT_CNT is a 32-bit register.

WT_CNT=0x4000_2008

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			CNT
-			0x000
.			RW

11 0	CNT	Watch Timer Counter bits.

11. 16-bit timer

16-bit timer block comprises 4 channels of 16-bit general purpose timers. Each channel has an independent 16-bit counter and a dedicated prescaler that feeds a counting clock. 16-bit timer supports periodic timer, PWM pulse, one-shot and capture mode. In addition, one more optional free-run timer is provided. Main purpose of 16-bit timer is a periodical tick timer.

16-bit timer of A31G32x series features the followings:

- 16-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 12-bit prescaler
- Synchronous start and clear function

Table 46 introduces pins assigned for 16-bit timer.

Table 46. Pin Assignment of 16-bit Timer: External Pins

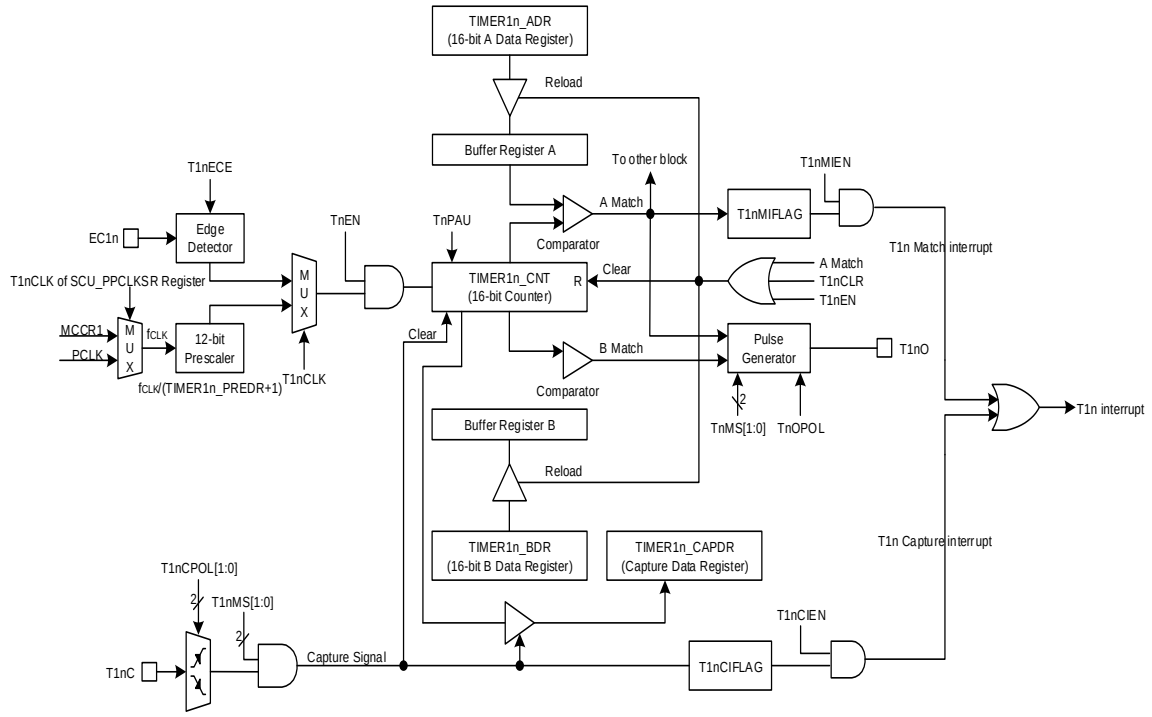
Pin name	Type	Description
EC1n	I	Timer 1n External Clock input
T1nCAP	I	Timer 1n Capture input
T1nOUT	O	Timer 1n Output

NOTE:

1. n = 0, 1, 2, and 3

11.1 16-bit timer block diagram

In this section, 16-bit timer is described in a block diagram in Figure 66.



NOTES:

1. The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the TnEN bit is set, the counting of TIMERn_CNT may be delayed 2 clocks or more.

Figure 66. 16-bit Timer Block Diagram

11.2 Registers

Base address of 16-bit timer is introduced in the followings:

Table 47. Base Address of 16-bit Timer

Name	Base address
TIMER10	0x4000_2100
TIMER11	0x4000_2200
TIMER12	0x4000_2300
TIMER13	0x4000_2700

Table 48. TIMER 1n Register Map

Name	Offset	Type	Description	Reset value	Reference
TIMER1n_CR	0x0000	RW	Timer/Counter 1n Control Register	0x0000_0000	11.2.1
TIMER1n_ADR	0x0004	RW	Timer/Counter 1n A Data Register	0x0000_FFFF	11.2.2
TIMER1n_BDR	0x0008	RW	Timer/Counter 1n B Data Register	0x0000_FFFF	11.2.3
TIMER1n_CAPDR	0x000C	RO	Timer/Counter 1n Capture Data Register	0x0000_0000	11.2.4
TIMER1n_PREDR	0x0010	RW	Timer/Counter 1n Prescaler Data Register	0x0000_0FFF	11.2.5
TIMER1n_CNT	0x0014	RO	Timer/Counter 1n Counter Register	0x0000_0000	11.2.6

NOTE:

1. n = 0, 1, 2, and 3

11.2.1 TIMER1n_CR: Timer/counter 1n control register

Timer module should be configured properly before running. When a target purpose is defined, the timer can be configured in the TIMERn_CR register. After configuring TIMER1n_CR, a user can start or stop the timer function by using TIMERn_CR.

TIMER1n_CR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0, 1, 2, and 3)

TIMER10_CR=0x4000_2100, TIMER11_CR=0x4000_2200
TIMER12_CR=0x4000_2300, TIMER13_CR=0x4000_2700

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		TnEN	TnCLK	TnMS		TnECE	Reserved		TnOPOL	TnCPOL		TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
-		0	0	00		0	-		0	00		0	0	0	0	0	0
-		RW	RW	RW		RW	-		RW	RW		RW	RW	RW	RW	RW	RW

15	T1nEN	Timer 1n Operation Enable bit.	
		0	Disable timer 1n operation.
		1	Enable timer 1n operation. (Counter clear and start)
14	T1nCLK	Timer 1n Clock Selection bit.	
		0	Select an internal prescaler clock.
		1	Select an external clock.
		NOTE: This bit should be changed during T1nEN bit is "0b".	

13 12	T1nMS	Timer 1n Operation Mode Selection bits.	
		00	Timer/Counter mode. (T1nO: Toggle at A-match)
		01	Capture mode. (The A-match interrupt can occur)
		10	PPG one-shot mode. (T1nOUT: Programmable pulse output)
		11	PPG repeat mode. (T1nOUT: Programmable pulse output)
		Note) This bit should be changed during TnEN bit is "0b".	
11	T1nECE	Timer 1n External Clock Edge Selection bit.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.
8	T1nOPOL	T1nOUT Polarity Selection bit.	
		0	Start high. (T1nOUT is low level at disable)
		1	Start low. (T1nOUT is high level at disable)
7 6	T1nCPOL	Timer 1n Capture Polarity Selection bits.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both of falling and rising edge.
		11	Reserved.
5	T1nMIEN	Timer 1n Match Interrupt Enable bit.	
		0	Disable timer 1n match interrupt.
		1	Enable timer 1n match interrupt.
4	T1nCIEN	Timer 1n Capture Interrupt Enable bit.	
		0	Disable timer 1n capture interrupt.
		1	Enable timer 1n capture interrupt.
3	T1nMIFLAG	Timer 1n Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
2	T1nCIFLAG	Timer 1n Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
1	T1nPAU	Timer 1n Counter Temporary Pause Control bit.	
		0	Continue counting.
		1	Temporary pause.
0	T1nCLR	Timer 1n Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer 1n counter and prescaler. (Automatically cleared to "0b" after operation)

11.2.2 TIMER1n_ADR: Timer/counter 1n A data register

TIMER1n_ADR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0, 1, 2, and 3)

TIMER10_ADR=0x4000_2104, TIMER11_ADR =0x4000_2204
TIMER12_ADR =0x4000_2304, TIMER13_ADR =0x4000_2704

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		ADATA	
-		0xFFFF	
-		RW	

15 0	ADATA	Timer/Counter 1n A Data bits. The range is 0x0002 to 0xFFFF. **NOTE:** 1. Do not write "0000H" in the TIMER1n_ADR register when PPG mode.

11.2.3 TIMER1n_BDR: Timer/counter n B data register

TIMER1n_BDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0, 1, 2, and 3)

TIMER10_BDR=0x4000_2108, TIMER11_BDR =0x4000_2208
TIMER12_BDR =0x4000_2308, TIMER13_BDR =0x4000_2708

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		BDATA	
-		0xFFFF	
-		RW	

15 0	BDATA	Timer/Counter 1n B Data bits. The range is 0x0000 to 0xFFFF.

11.2.4 TIMER1n_CAPDR: Timer/counter 1n capture data register

TIMER1n_ CAPDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_CAPDR =0x4000_210C, TIMER11_CAPDR =0x4000_220C
TIMER12_CAPDR =0x4000_230C, TIMER13_CAPDR =0x4000_270C

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		CAPD	
-		0x0000	
-		RO	

15 0	CAPD	Timer/Counter 1n Capture Data bits.

11.2.5 TIMER1n_PREDR: Timer/counter 1n prescaler data register

TIMER1n_PREDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0, 1, 2, and 3)

TIMER10_PREDR =0x4000_2110, TIMER11_PREDR =0x4000_2210
TIMER12_PREDR =0x4000_2310, TIMER13_PREDR =0x4000_2710

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		PRED	
-		0xFFF	
-		RW	

11 0	PRED	Timer/Counter 1n Prescaler Data bits.

11.2.6 TIMER1n_CNT: Timer/counter n counter register

TIMER1n_CNT is a 32-bit register, and able to do 32/16/8-bit access. (n = 0, 1, 2, and 3)

TIMER10_CNT =0x4000_2114, TIMER11_CNT =0x4000_2214
TIMER12_CNT =0x4000_2314, TIMER13_CNT =0x4000_2714

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		CNT	
-		0x0000	
-		RO	

15 0	CNT	Timer/Counter 1n Counter bits.

11.3 Functional description

11.3.1 Timer counter 10/11/12/13

Timer/counter n can be clocked by an internal or an external clock source (EC1n). Its clock source is selected by a clock selection logic which is controlled by clock selection bits (T1nCLK). (n = 0, 1, 2, and 3)

- TIMER 1n clock source: fCLK/(TIMER1n_PREDR +1), EC1n

In capture mode, by TnCAP, data is captured into input capture data register (TIMER1n_CAPDR). TIMER 1n results the comparison between a counter and the data register through T1nOUT port in timer/counter mode. In addition, TIMER 1n outputs PWM wave form through T1nOUT port in the PPG mode.

Table 49 introduces various operating modes of TIMER 1n according to the value of timer/counter register.

Table 49. TIMER 1n Operating Modes

T1nEN	Alternative mode	T1nMS[1:0]	TIMER1n_PREDR	Timer 1n
1	AF6	00	0xXXX	16-bit Timer/Counter Mode
1	AF5	01	0xXXX	16-bit Capture Mode
1	AF6	10	0xXXX	16-bit PPG Mode(one-shot mode)
1	AF6	11	0xXXX	16-bit PPG Mode(repeat mode)

11.3.2 16-bit timer/counter mode

16-bit timer/counter mode is selected by control register as shown in Figure 67. The 16-bit timer has a counter and data register. The counter register is increased by internal or external clock input. Timer n can use the clock input with 12-bit prescaler division rates (TIMER1n_PREDR) and external clock (EC1n). When each value of TIMER1n_CNT and TIMERn_ADR are identical in timer 1n, a match signal is generated and the interrupt of Timer 1n occurs. The TIMER1n_CNT values are automatically cleared by the match signal. It can be also cleared by software (T1nCLR).

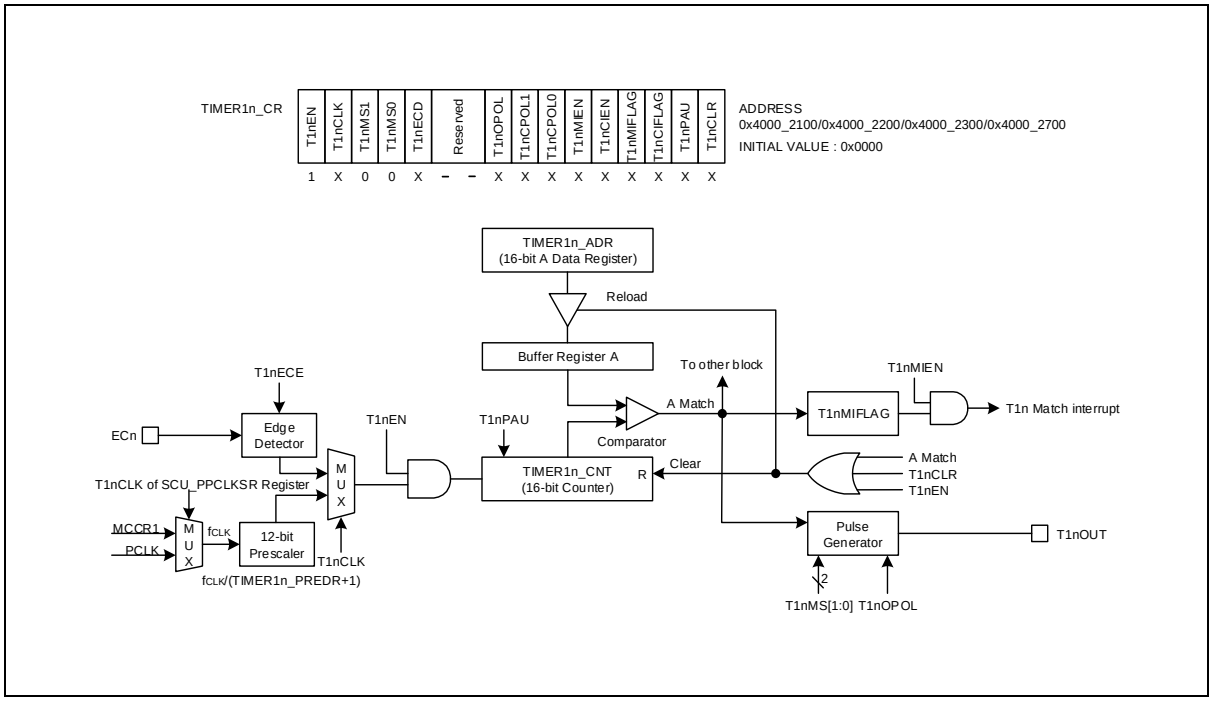


Figure 67. TIMER 1n Block Diagram in Timer/Counter Mode (n = 0, 1, 2, and 3)

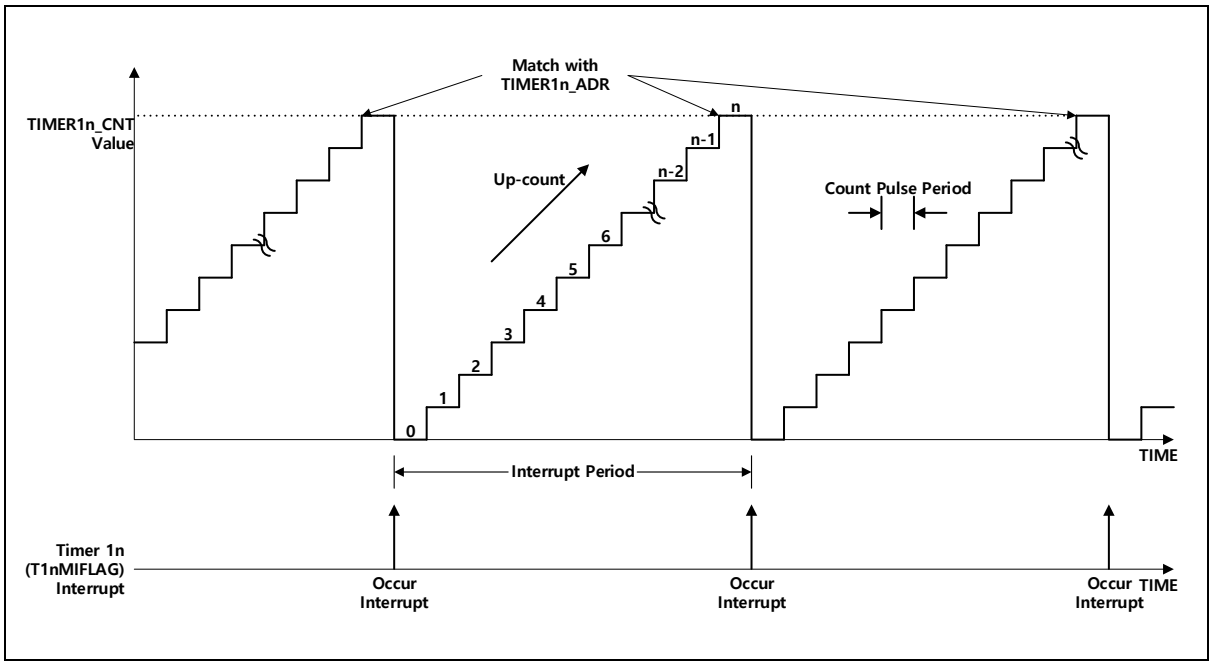


Figure 68. Timer/Counter Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3)

Figure 69 shows the timer/counter mode operation of timer/counter 1n. Refer to Figure 13 for internal timer clock source and Table 25 for Timer1n Output pin.

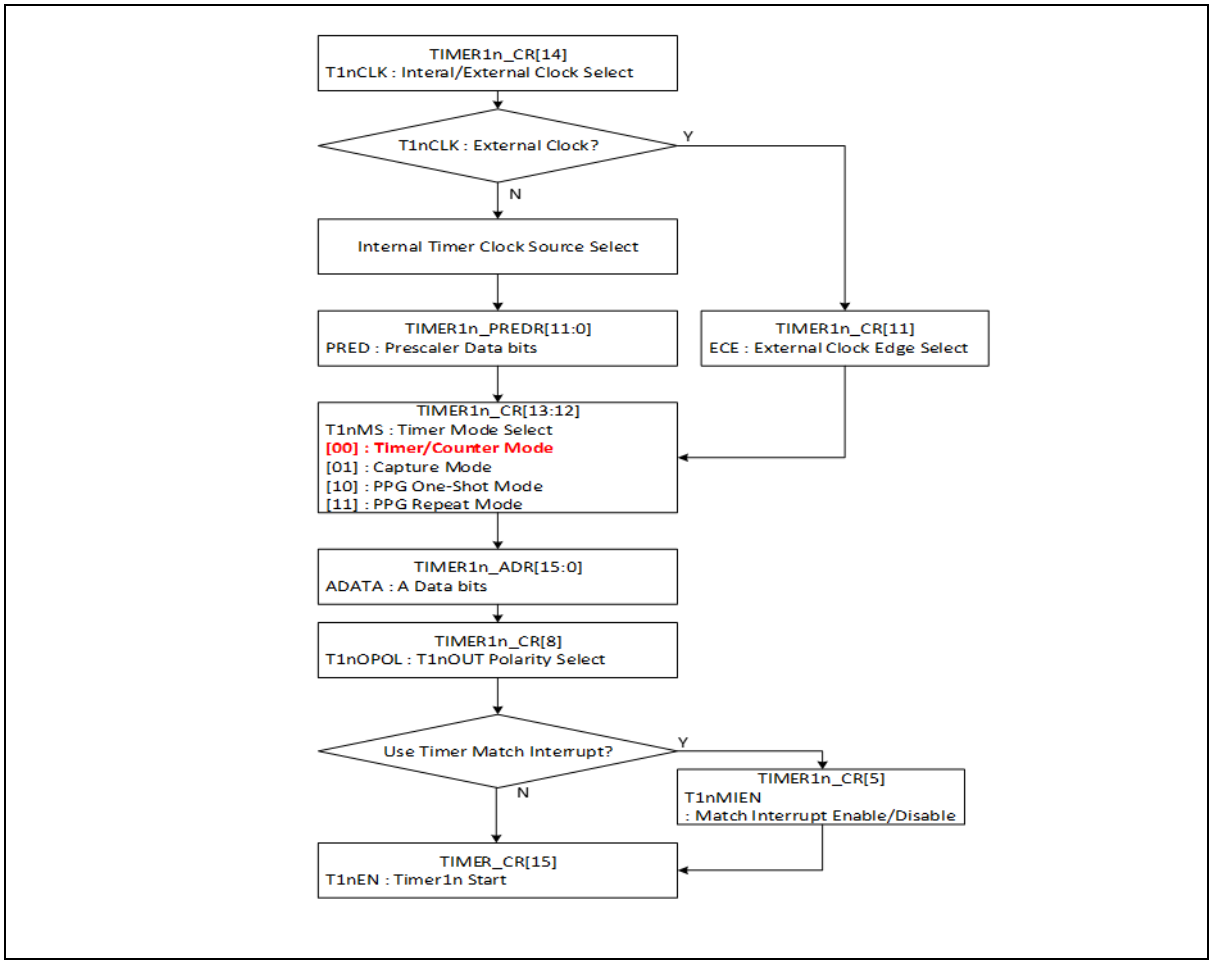


Figure 69. Timer/Counter Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3)

11.3.3 16-bit capture mode

Timer n capture mode is evoked by setting T1nMS[1:0] as '01'. It can use an internal clock as a clock source. Basically, it has the same function as 16-bit timer/counter mode, and the interrupt occurs when TIMER1n_CNT is equal to TIMER1n_ADR. TIMER1n_CNT value can be automatically cleared by a match signal. It can be cleared by software too (TnCLR).

A timer interrupt in capture mode is very useful when pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER1n_CAPDR. In the timer n capture mode, timer 1n output (TnO) waveform is not available.

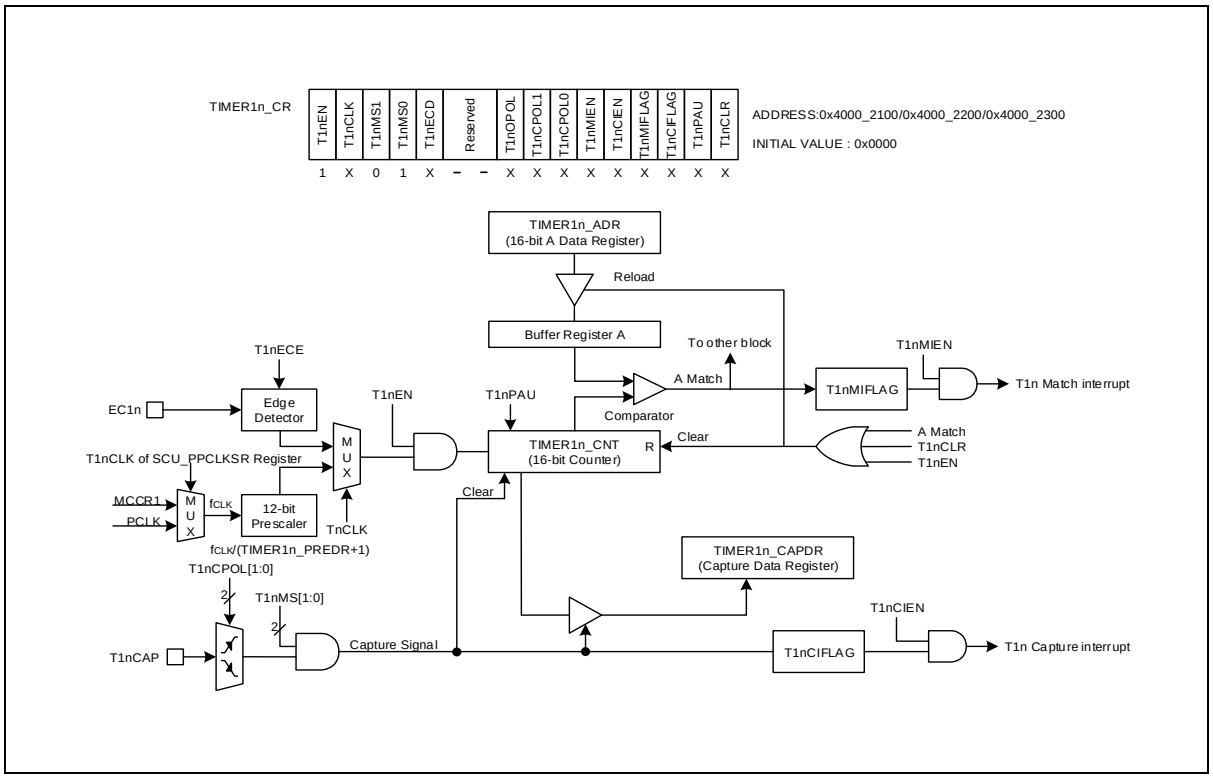


Figure 70. TIMER 1n Block Diagram in Capture Mode (n = 0, 1, 2, and 3)

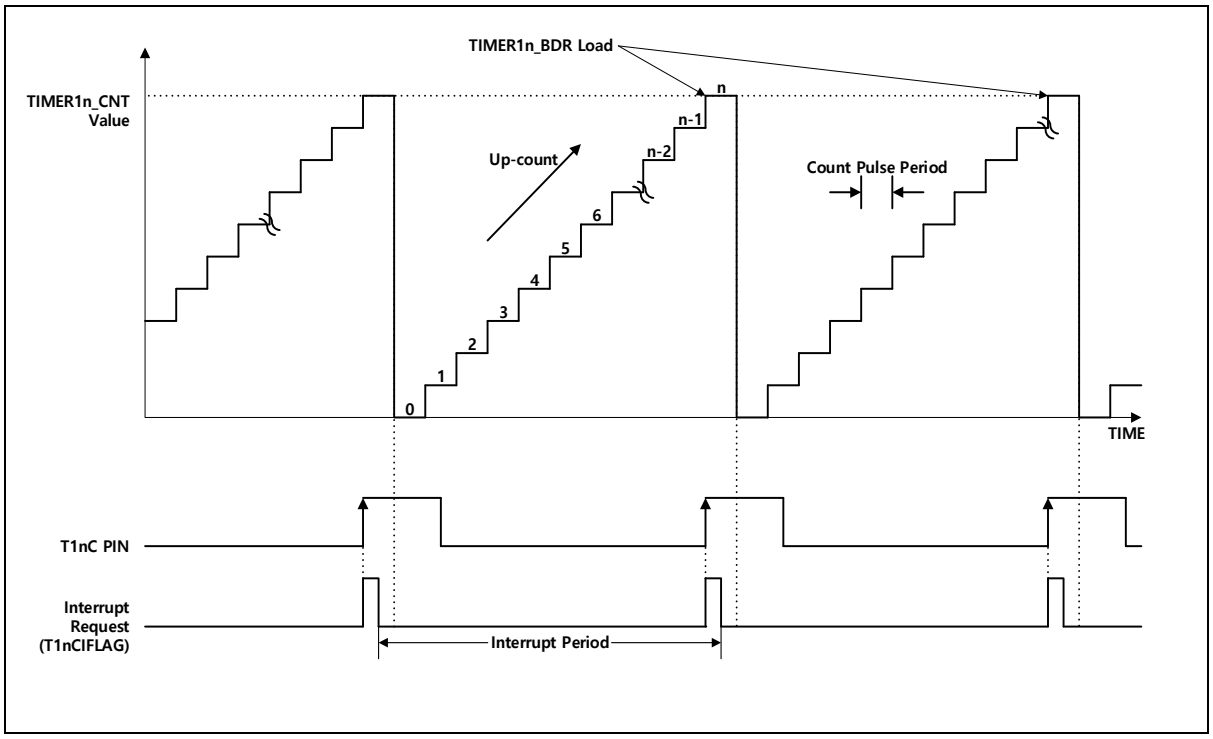


Figure 71. Capture Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3)

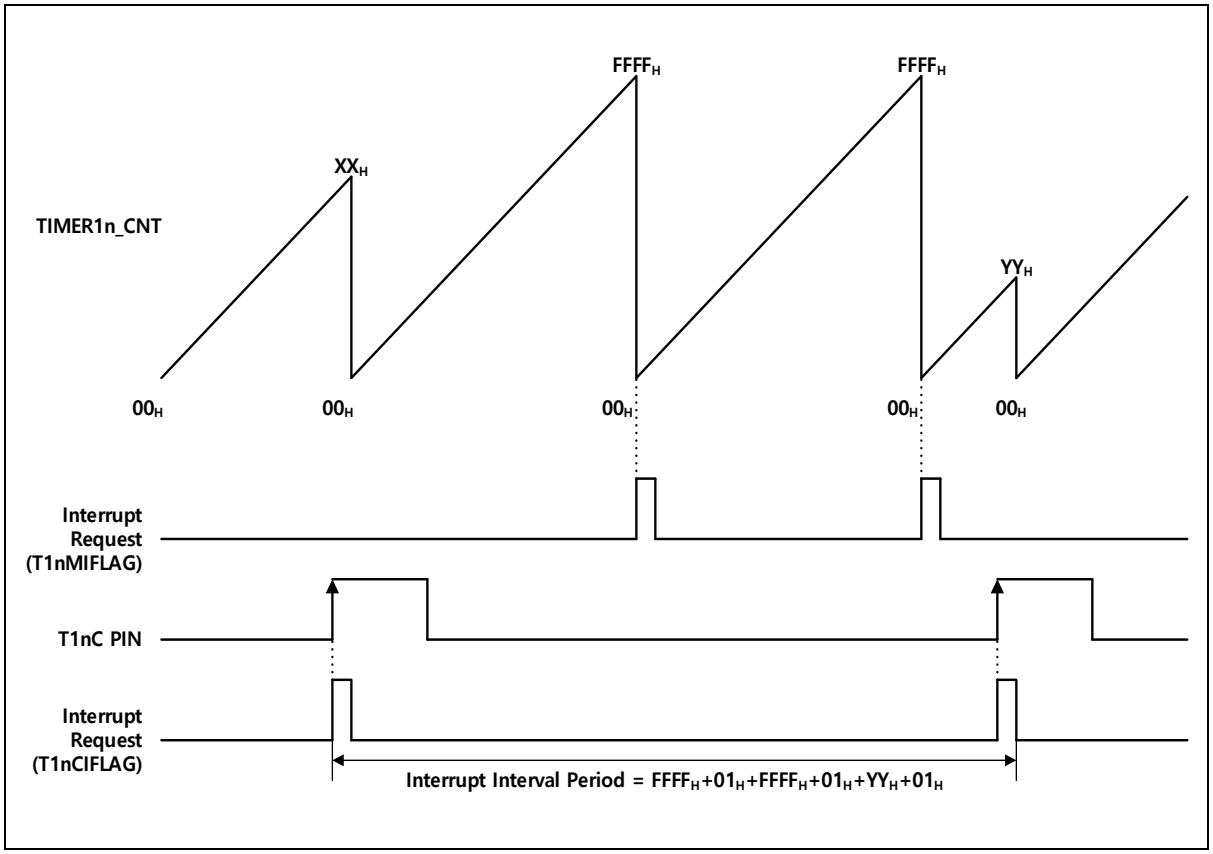


Figure 72. Express Timer Overflow in Capture Mode of TIMER 1n (n = 0, 1, 2, and 3)

Figure 73 shows the capture mode operation of timer/counter 1n. Refer to Figure 13 for internal timer clock source and section Table 25 for Timer1n Output pin.

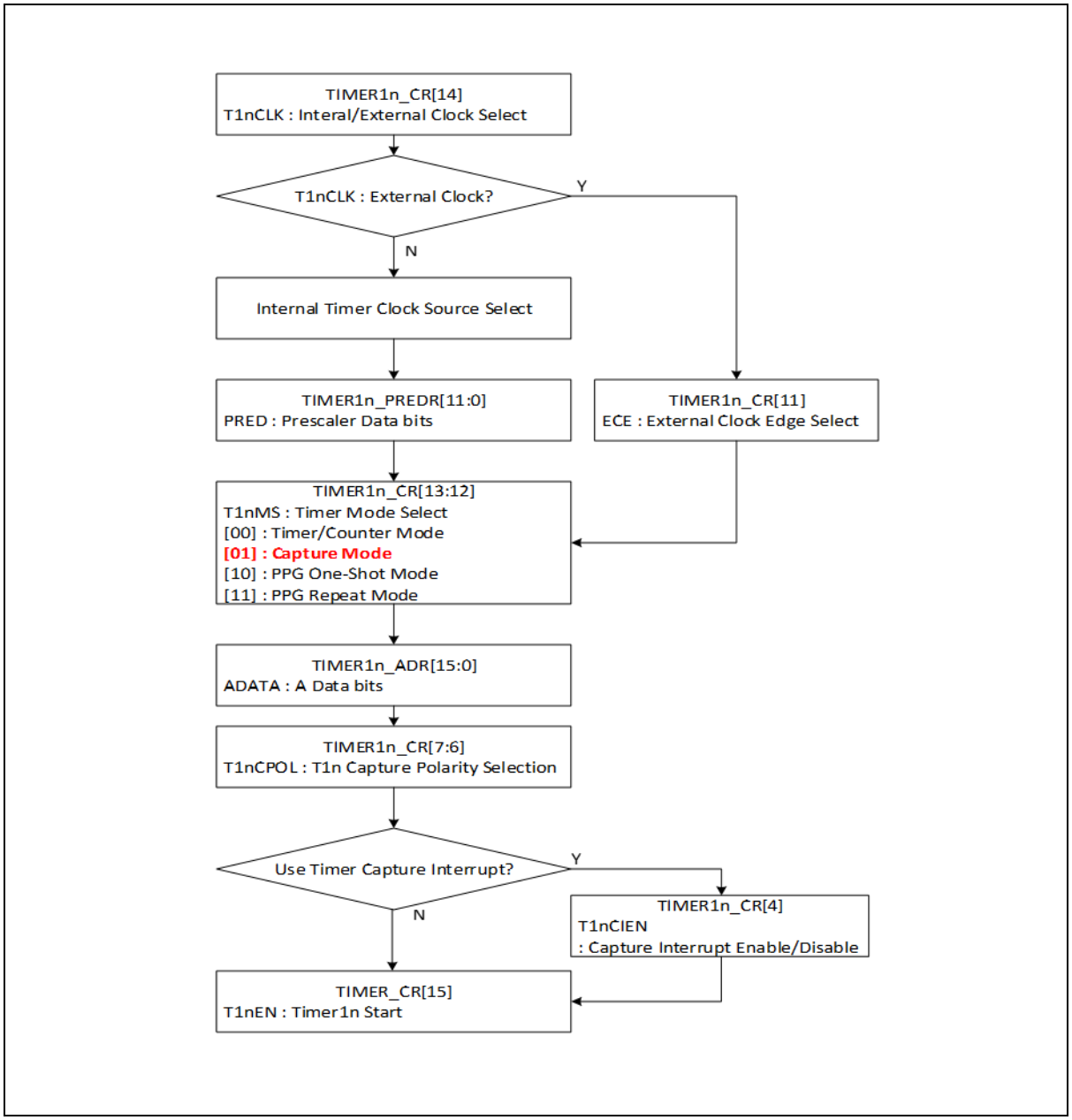
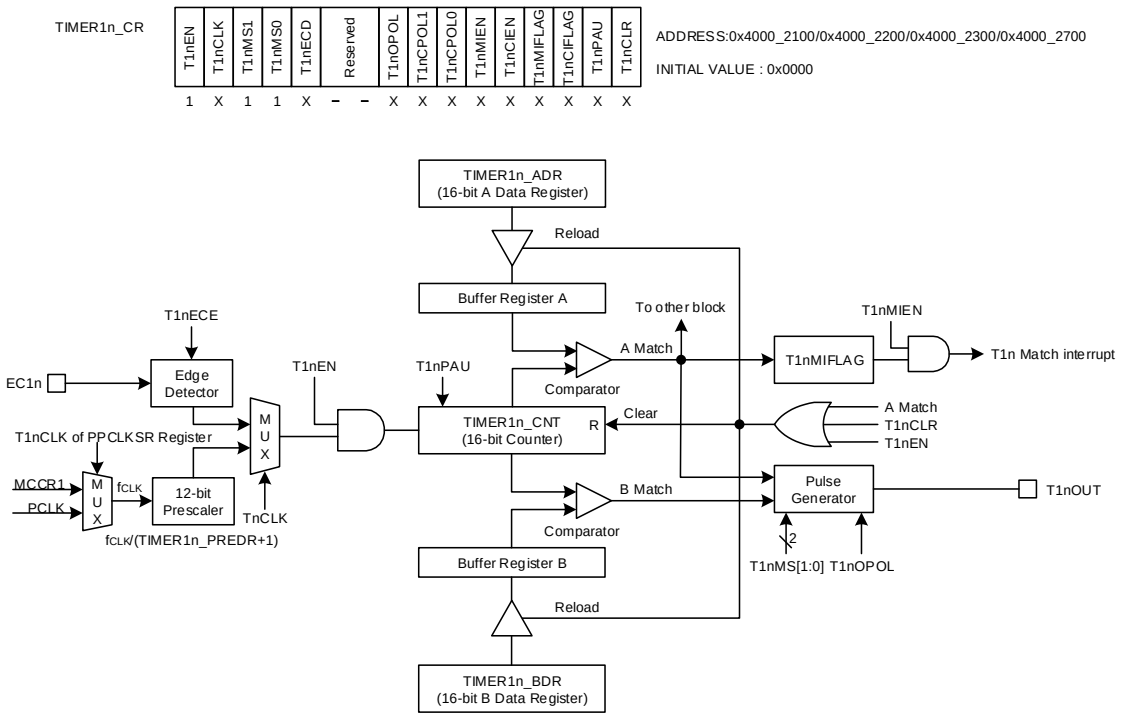


Figure 73. Capture Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3)

11.3.4 16-bit PPG mode

Timer 1n has a PPG (Programmable Pulse Generation) function. In PPG mode, the TnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding PnAFSRx to 'AF6'. Period of the PWM output is determined by the TIMERn_ADR, and duty of the PWM output is determined by the TIMER1n_BDR.



NOTE:

1. The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 74. TIMER 1n Block Diagram in PPG Mode (n = 0, 1, 2, and 3)

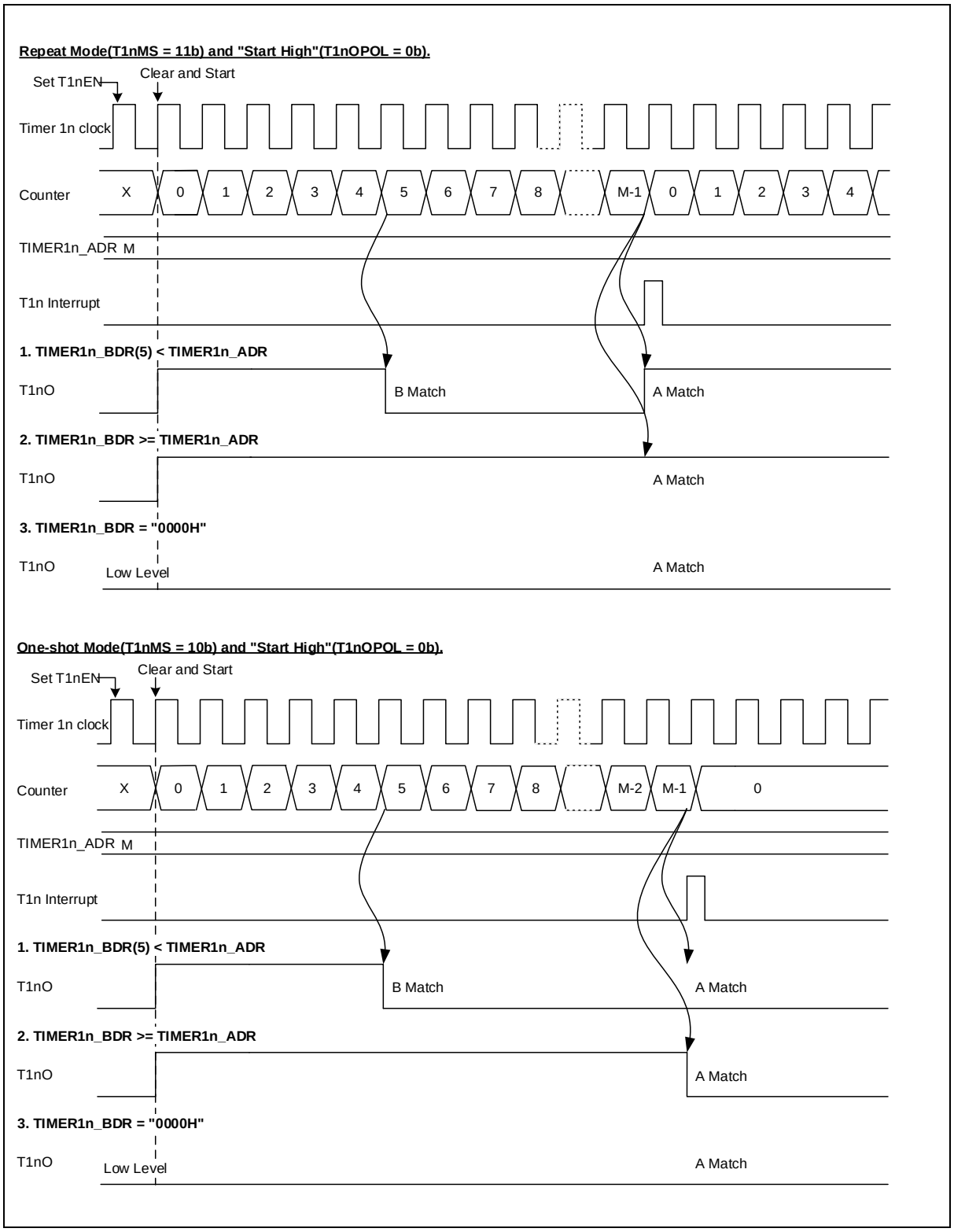


Figure 75. PPG Mode Timing Example of TIMER 1n (n = 0, 1, 2, and 3)

Figure 76 shows the PPG mode operation of timer/counter 1n. Refer to Figure 13 for internal timer clock source and Table 25 for Timer1n Output pin.

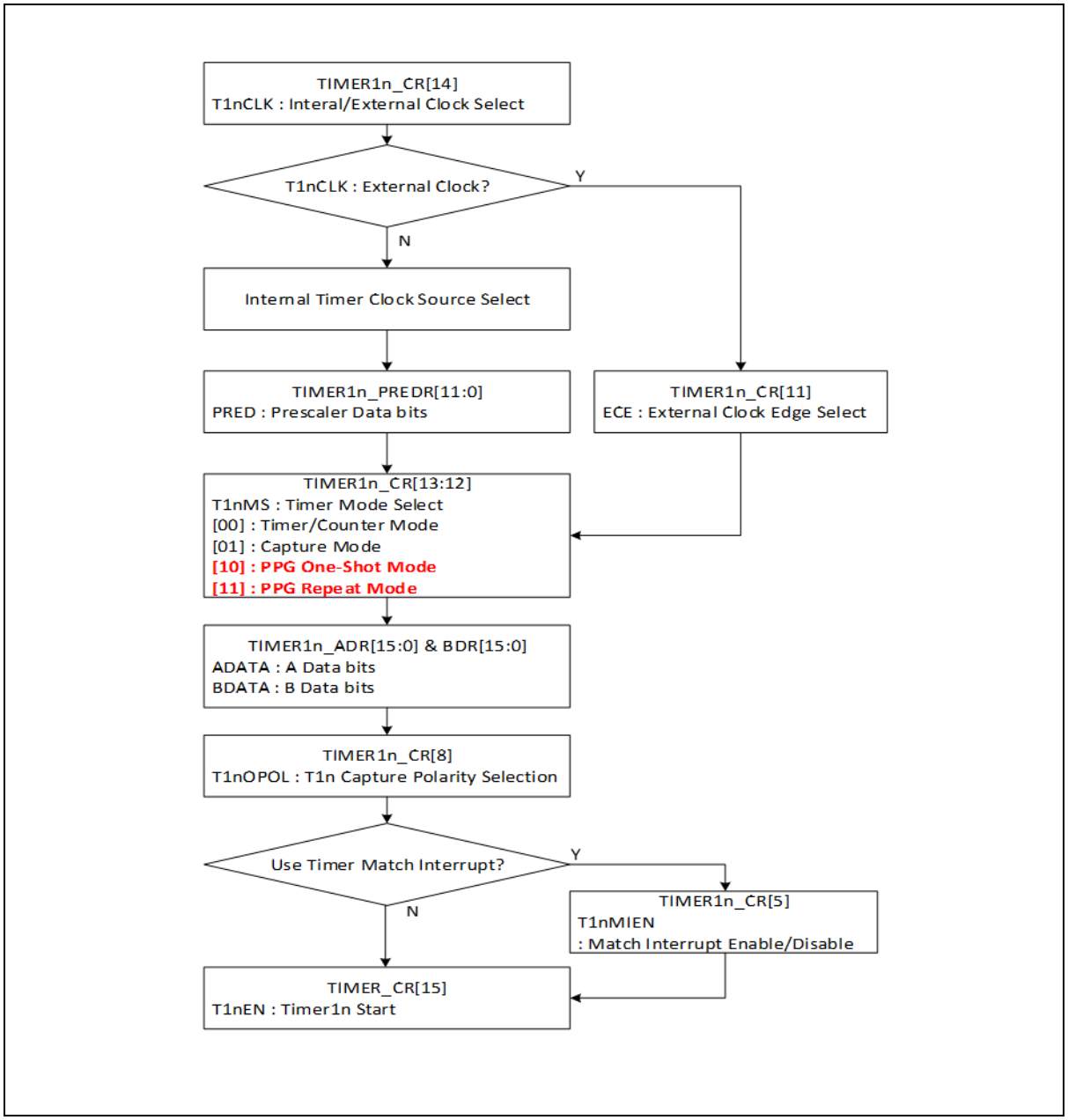


Figure 76. PPG Mode Operation Sequence of TIMER 1n (n = 0, 1, 2, and 3)

12. 32-bit timer

32-bit timer block comprises 2 channels of 32-bit general purpose timers. Each channel has an independent 32-bit counter and a dedicated prescaler that feeds a counting clock. 32-bit timer supports periodic timer, PWM pulse, one-shot and capture mode. In addition, one more optional free-run timer is provided. Main purpose of 32-bit timer is a periodical tick timer.

32-bit timer of A31G32x series features the followings:

- 32-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 12-bit prescaler
- Synchronous start and clear function

Table 50 introduces pins assigned for 32-bit timer.

Table 50. Pin Assignment of 32-bit Timer: External Pins

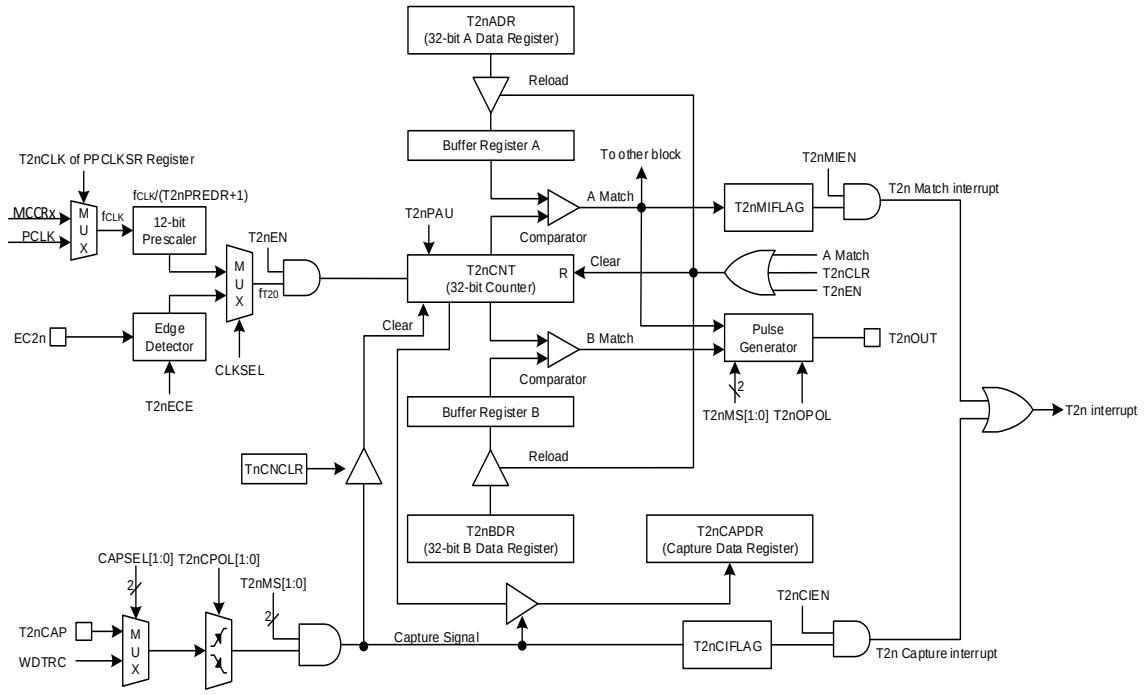
Pin name	Type	Description
EC2n	I	Timer 2n external clock input
T2nCAP	I	Timer 2n capture input
T2nOUT	O	Timer/PWM/one-shot output

NOTE:

1. n = 0 or 1

12.1 32-bit timer block diagram

In this section, 32-bit timer is described in a block diagram in Figure 77.



NOTES:

1. The T20EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the T20EN bit is set, the counting of TIMER20_CNT may be delayed 2 clocks or more.

Figure 77. 16-bit Timer Block Diagram

12.2 Registers

Base address of 32-bit timer is introduced in the followings:

Table 51. Base Address of 32-bit Timer

Name	Base address
TIMER20	0x4000_2500
TIMER21	0x4000_2600

Table 52. TIMER 2n Register Map

Name	Offset	Type	Description	Reset value	Reference
TIMER2n_CR	0x0000	RW	Timer/Counter 2n Control Register	0x0000_0000	12.2.1
TIMER2n_ADR	0x0004	RW	Timer/Counter 2n A Data Register	0xFFFF_FFFF	12.2.2
TIMER2n_BDR	0x0008	RW	Timer/Counter 2n B Data Register	0xFFFF_FFFF	12.2.3
TIMER2n_CAPDR	0x000C	RO	Timer/Counter 2n Capture Data Register	0x0000_0000	12.2.4
TIMER2n_PREDR	0x0010	RW	Timer/Counter 2n Prescaler Data Register	0x0000_0FFF	12.2.5
TIMER2n_CNT	0x0014	RO	Timer/Counter 2n Counter Register	0x0000_0000	12.2.6

NOTE:

1. n = 0 or 1

12.2.1 TIMER2n_CR: Timer/counter 2n control register

Timer module should be configured properly before running. When a target purpose is defined, the timer can be configured in the TIMER2n_CR register. After configuring TIMER2n_CR, a user can start or stop the timer function by using TIMER2n_CR.

TIMER1n_CR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0 or 1)

TIMER20_CR=0x4000_2500, TIMER21_CR=0x4000_2600

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		T2nCNCLR	T20EN	T20CLK	T20MS		T20ECE	CAPSEL		T20OPOL	T20CPOL		T20MIEN	T20CIEN	T20MIFLAG	T20CIFLAG	T20PAU	T20CLR
-			0	0	0			0			0		0	0	0	0	0	0
-		RW	RW	RW	RW		RW	RW		RW	RW		RW	RW	RW	RW	RW	RW

16	T2nCNCLR	Timer 2n counter clear after Capture	
		0	Clear counter after capture
		1	Keep counter after capture
15	T2nEN	Timer 2n Operation Enable bit.	
		0	Disable timer 2n operation.
		1	Enable timer 2n operation. (Counter clear and start)
14	T2nCLK	Timer 2n Clock Selection bit.	
		0	Select an internal prescaler clock.
		1	Select an external clock.
NOTES: This bit should be changed during T2nEN bit is "0b". If you select an internal prescaler clock, you should set T2nCLK bit in the SCU_PPCLKSR register first.			
13 12	T2nMS	Timer 2n Operation Mode Selection bits.	
		00	Timer/Counter mode. (T2nO: Toggle at A-match)
		01	Capture mode. (The A-match interrupt can occur)
		10	PPG one-shot mode. (T2nO: Programmable pulse output)
		11	PPG repeat mode. (T2nO: Programmable pulse output)
		NOTE: This bit should be changed during T2nEN bit is "0b".	
11	T2nECE	Timer 2n External Clock Edge Selection bit.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.
10 9	CAPSEL	Timer 2n Capture Signal Selection bits.	
		00	Select an external capture signal.
		01	Not used
		10	Select the WDTRC (Watch-dog timer RC oscillator) signal.
		11	Not used
		NOTE: This bit should be changed during T2nEN bit is "0b".	
8	T2nOPOL	T2nO Polarity Selection bit.	
		0	Start high. (T2nO is low level at disable)
		1	Start low. (T2nO is high level at disable)

7 6	T2nCPOL	Timer 2n Capture Polarity Selection bits.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both of falling and rising edge.
		11	Reserved.
5	T2nMIEN	Timer 2n Match Interrupt Enable bit.	
		0	Disable timer 2n match interrupt.
		1	Enable timer 2n match interrupt.
4	T2nCIEN	Timer 2n Capture Interrupt Enable bit.	
		0	Disable timer 2n capture interrupt.
		1	Enable timer 2n capture interrupt.
3	T2nMIFLAG	Timer 2n Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
2	T2nCIFLAG	Timer 2n Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
1	T2nPAU	Timer 2n Counter Temporary Pause Control bit.	
		0	Continue counting.
		1	Temporary pause.
0	T2nCLR	Timer 2n Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer 2n counter and prescaler. (Automatically cleared to "0b" after operation)

12.2.2 TIMER2n_ADR: Timer/counter 2n A data register

TIMER2n_ADR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0 or 1)

TIMER20_ADR=0x4000_2504, TIMER21_ADR=0x4000_2604

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
ADATA			
0xFFFFFFFF			
RW			

31 0	ADATA	Timer/Counter 2n A Data bits. The range is 0x0002 to 0xFFFFFFFF.
NOTE: Do not write "0000H" in the TIMER20_ADR register when PPG mode.		

12.2.3 TIMER2n_BDR: Timer/counter 2n B data register

TIMER2n_BDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 0 or 1)

TIMER20_BDR=0x4000_2508, TIMER21_BDR=0x4000_2608

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
BDATA			
0xFFFFFFFF			
RW			

31 0	BDATA	Timer/Counter 2n B Data bits. The range is 0x0000 to xFFFFFFFF.

12.2.4 TIMER2n_CAPDR: Timer/counter 2n capture data register

TIMER2n_ CAPDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_CAPDR=0x4000_250C, TIMER21_CAPDR=0x4000_260C

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
CAPD			
0x0000			
RO			

31 0	CAPD	Timer/Counter 2n Capture Data bits.

12.2.5 TIMER2n_PREDR: Timer/counter 2n prescaler data register

TIMER2n_PREDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_PREDR=0x4000_2510, TIMER21_PREDR=0x4000_2610

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			PRED
0x00000			0xFFF
-			RW

11 0	PRED	Timer/Counter 2n Prescaler Data bits. f_{CLK}/(TIMER2n_PREDR +1)

12.2.6 TIMER2n_CNT: Timer/counter 2n counter register

TIMER2n_CNT is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_CNT=0x4000_2514, TIMER21_CNT=0x4000_2514

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
CNT			
0x00000000			
RO			

31 0	CNT	Timer/Counter 2n Counter bits.

12.3 Functional description

12.3.1 Timer counter 2n

Timer/counter 2n can be clocked by an internal or an external clock source (EC2n). Its clock source is selected by a clock selection logic which is controlled by clock selection bits (T2nCLK).

- TIMER 2n clock source: fCLK/(TIMER2n_PREDR +1), EC2n

In capture mode, by T2nCAP, data is captured into input capture data register (TIMER2n_CAPDR). TIMER 2n results the comparison between a counter and the data register through T2nOUT port in timer/counter mode. In addition, TIMER 2n outputs PWM wave form through T2nOUT port in the PPG mode.

Table 53 introduces various operating modes of TIMER 2n according to the value of timer/counter register.

Table 53. TIMER 2n Operating Modes

T2nEN	Alternative mode	T2nMS[1:0]	TIMER2n_PREDR	TIMER 2n
1	AF6, AF8	00	0xXXX	32-bit Timer/Counter Mode
1	AF5, AF7	01	0xXXX	32-bit Capture Mode
1	AF6, AF8	10	0xXXX	32-bit PPG Mode(one-shot mode)
1	AF6, AF8	11	0xXXX	32-bit PPG Mode(repeat mode)

12.3.2 32-bit timer/counter mode

32-bit timer/counter mode is selected by control register as shown in Figure 78. The 32-bit timer has a counter and data register.

The counter register is increased by internal or external clock input. TIMER 2n can use the clock input with 12-bit prescaler division rates (TIMER2n_PREDR) and external clock (EC2n). When each value of TIMER20_CNT and TIMER2n_ADR are identical in TIMER 2n, a match signal is generated and the interrupt of Timer 2n occurs. The TIMER2n_CNT values are automatically cleared by the match signal. It can be also cleared by software (T2nCLR).

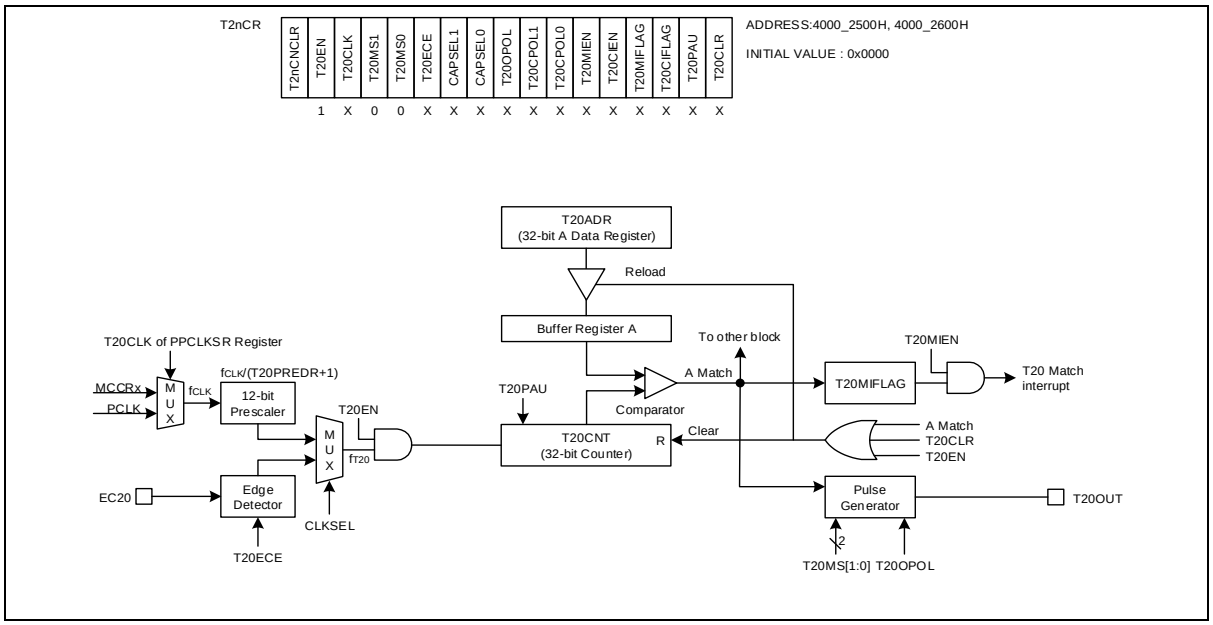


Figure 78. TIMER 2n Block Diagram in Timer/Counter Mode

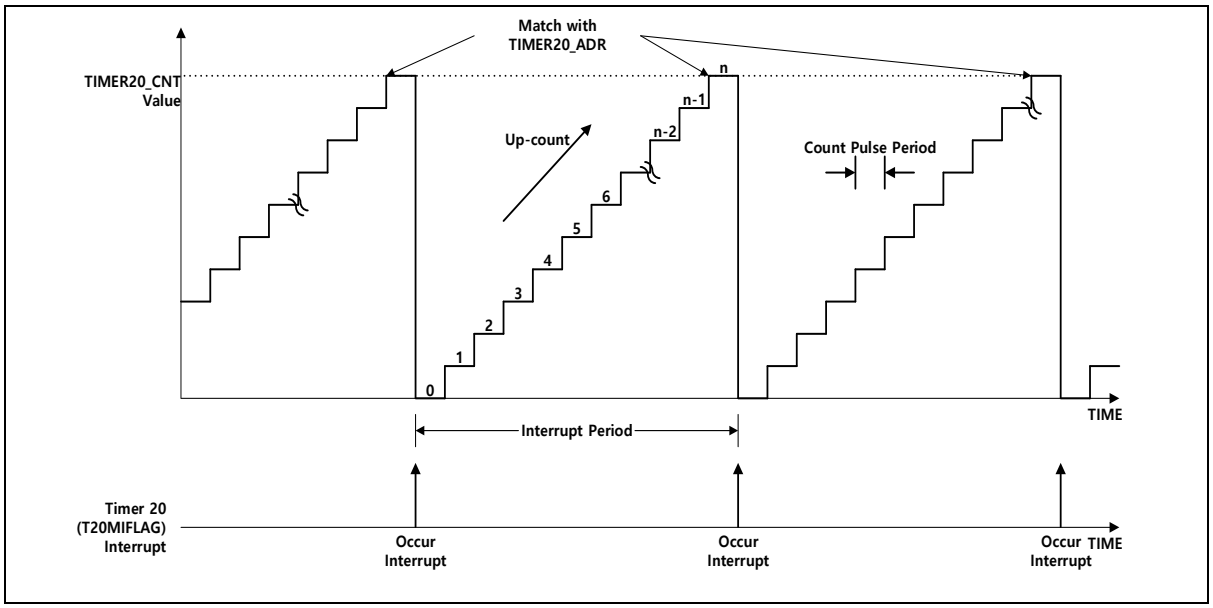


Figure 79. Timer/Counter Mode Timing Example of TIMER 2n

Figure 80 shows the timer/counter mode operation of timer/counter 2n. Refer to Figure 13 for internal timer clock source and Table 25 for Timer2n Output pin.

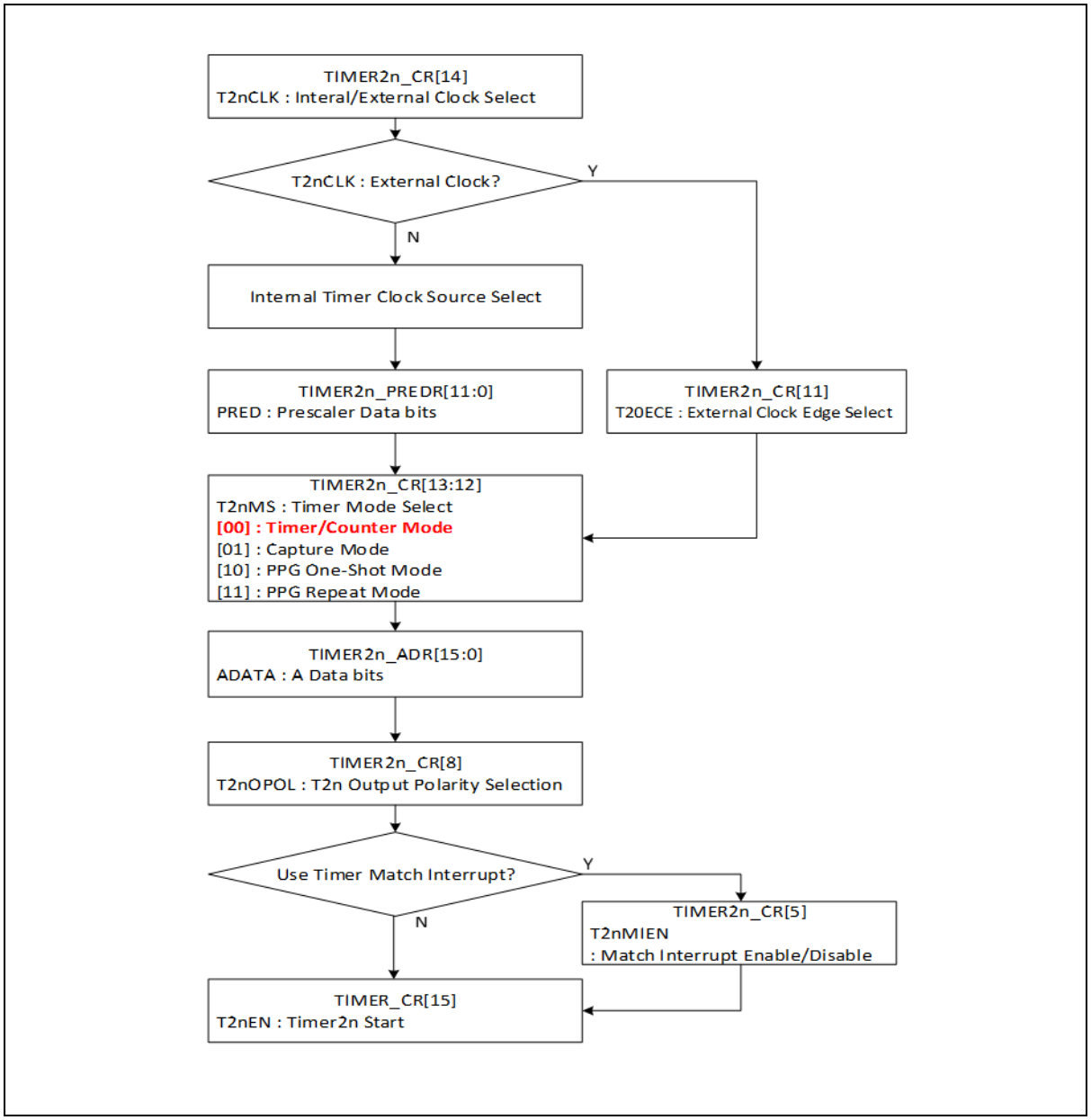


Figure 80. Timer/Counter Mode Operation Sequence of TIMER 2n (n = 0 or 1)

12.3.3 32-bit capture mode

Timer 2n capture mode is evoked by setting T2nMS[1:0] as '01'. It can use an internal clock as a clock source. Basically, it has the same function as 32-bit timer/counter mode, and the interrupt occurs when TIMER2n_CNT is equal to TIMER2n_ADR. TIMER2n_CNT value can be cleared by software (T2nCLR).

A timer interrupt in capture mode is very useful when pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER2n_BDR. In the TIMER 2n capture mode, TIMER 2n output (T2nOUT) waveform is not available. (n = 0 or 1).

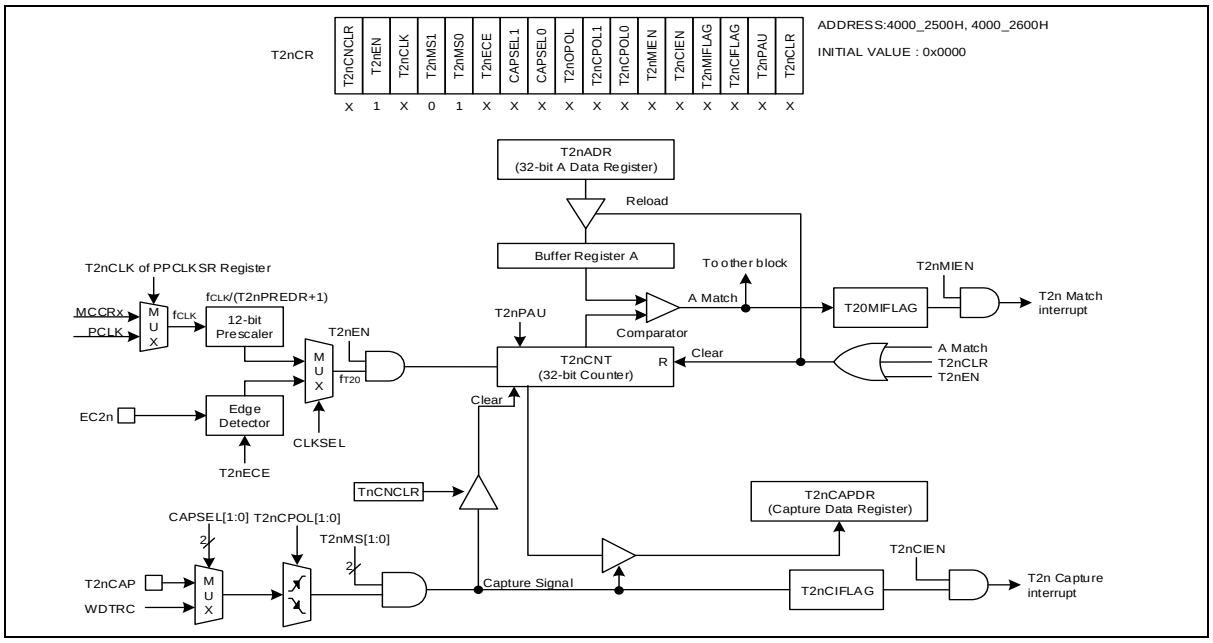


Figure 81. TIMER 2n Block Diagram in Capture Mode

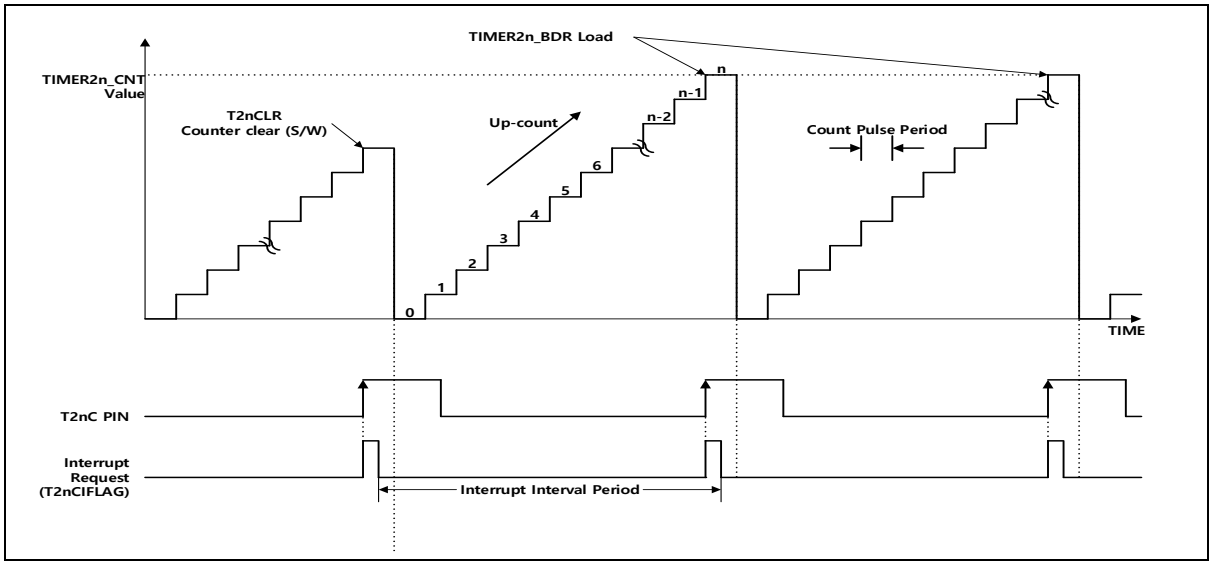


Figure 82. Capture Mode Timing Example of TIMER 2n (n = 0 and 1)

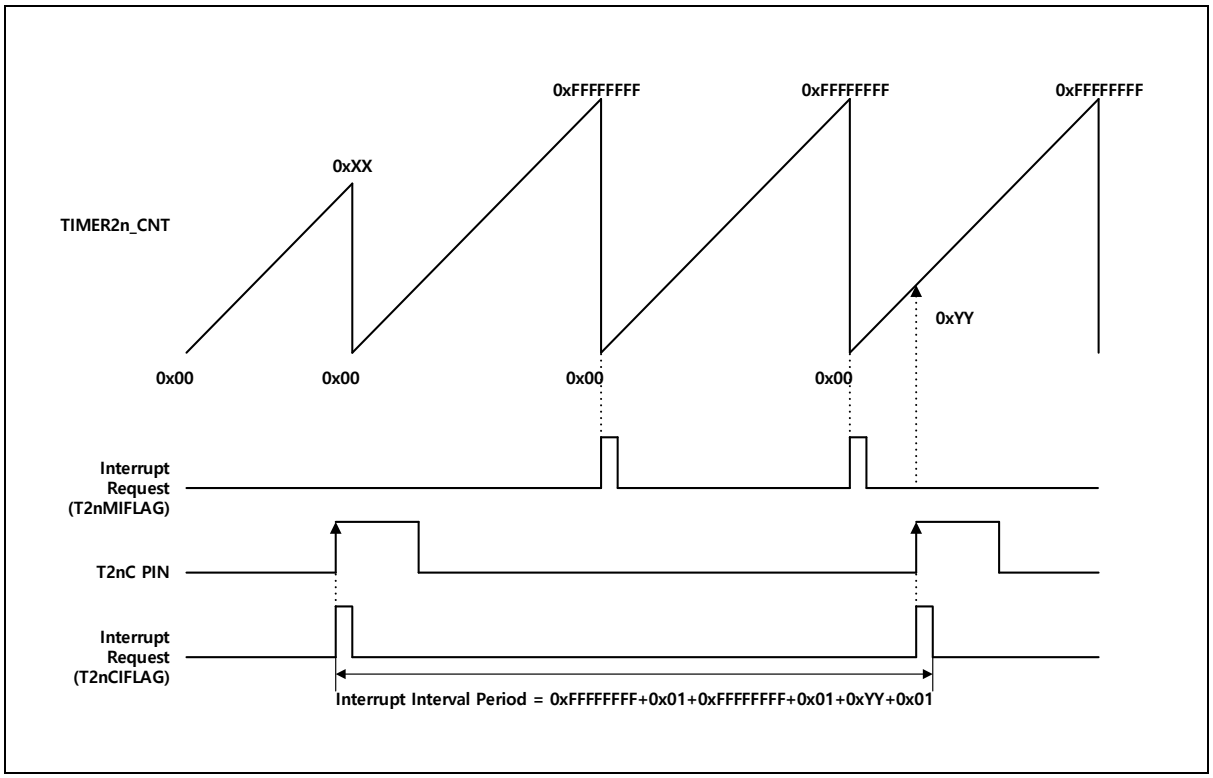


Figure 83. Express Timer Overflow in Capture Mode of TIMER 2n (T2nCNCLR = 1'b1)

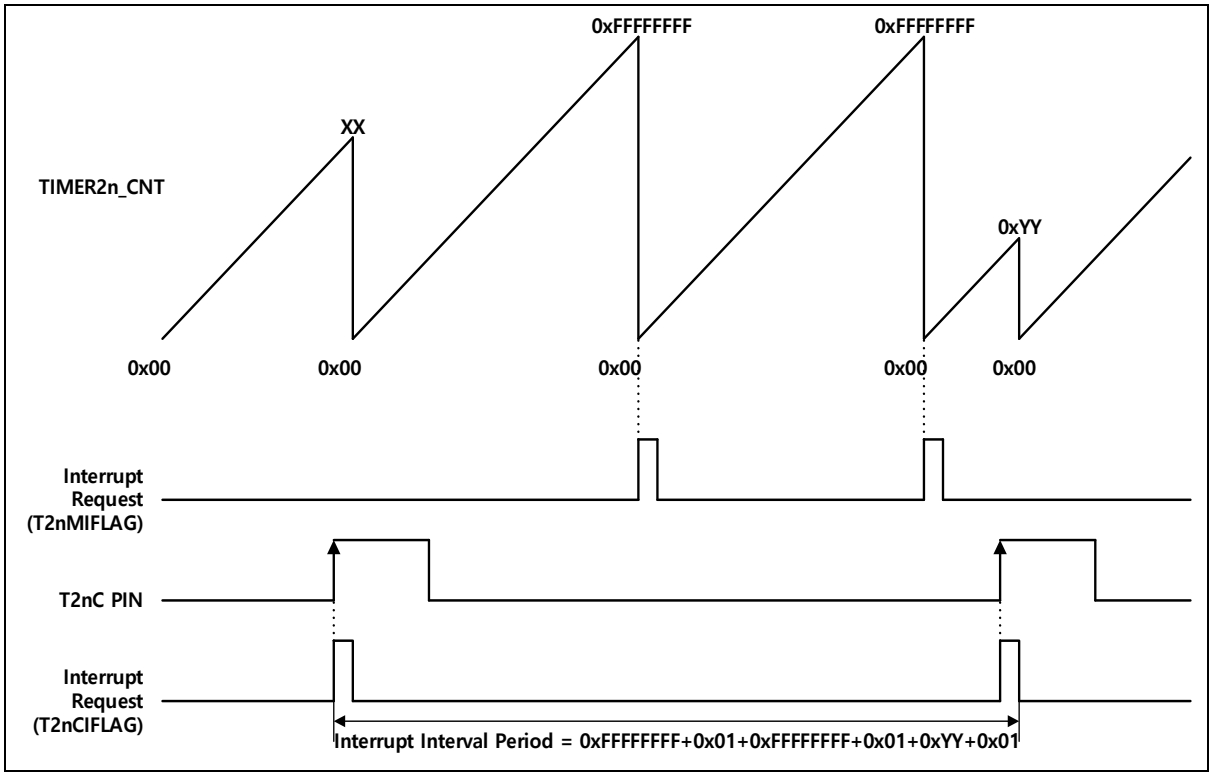


Figure 84. Express Timer Overflow in Capture Mode of TIMER 2n (T2nCNCLR = 1'b0)

Figure 85 shows the capture mode operation of timer/counter 2n. Refer to Figure 13 for internal timer clock source and Table 25 for Timer2n Output pin.

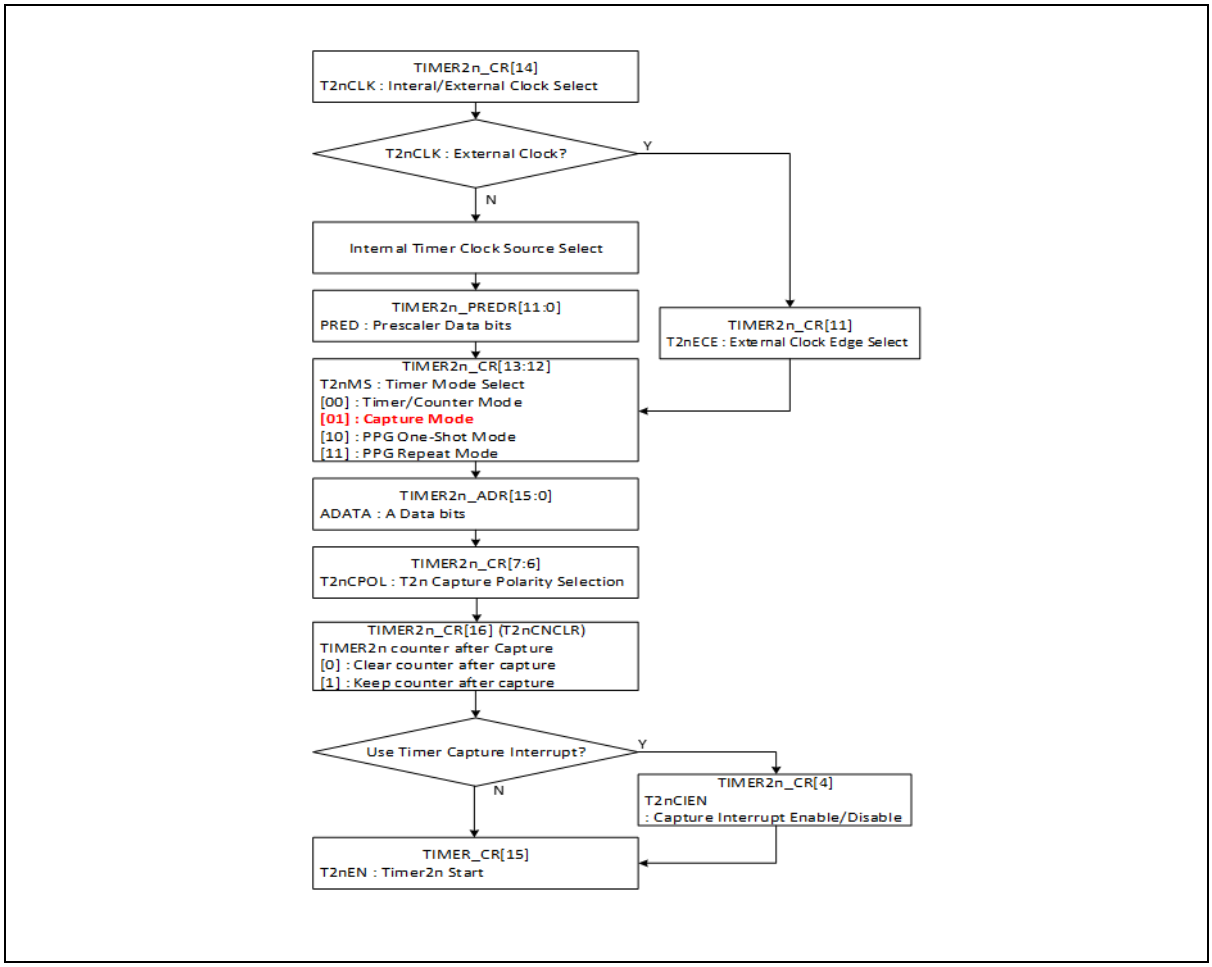
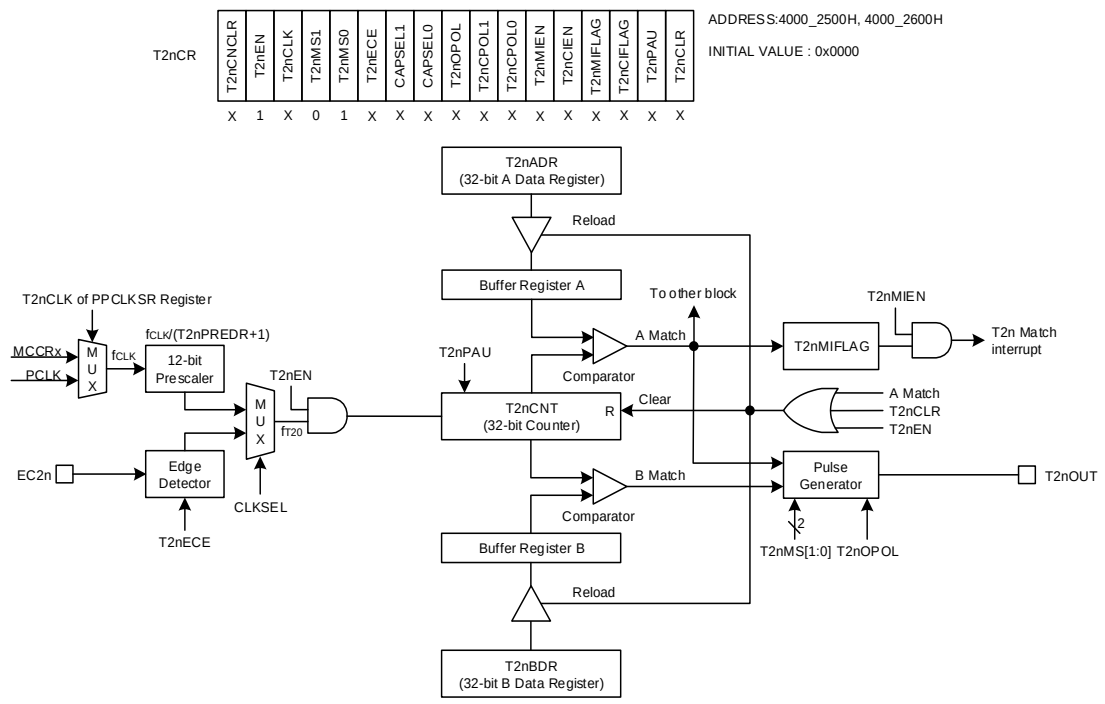


Figure 85. Capture Mode Operation Sequence of TIMER 2n (n = 0 or 1)

12.3.4 32-bit PPG mode

Timer 2n has a PPG (Programmable Pulse Generation) function. In PPG mode, the T2nOUT pin outputs up to 32-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding PnAFSRx to 'AF6' or 'AF7'. Period of the PWM output is determined by the TIMER2n_ADR, and duty of the PWM output is determined by the TIMER2n_BDR.



NOTE:

1. The T20EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 86. TIMER 2n Block Diagram in PPG Mode

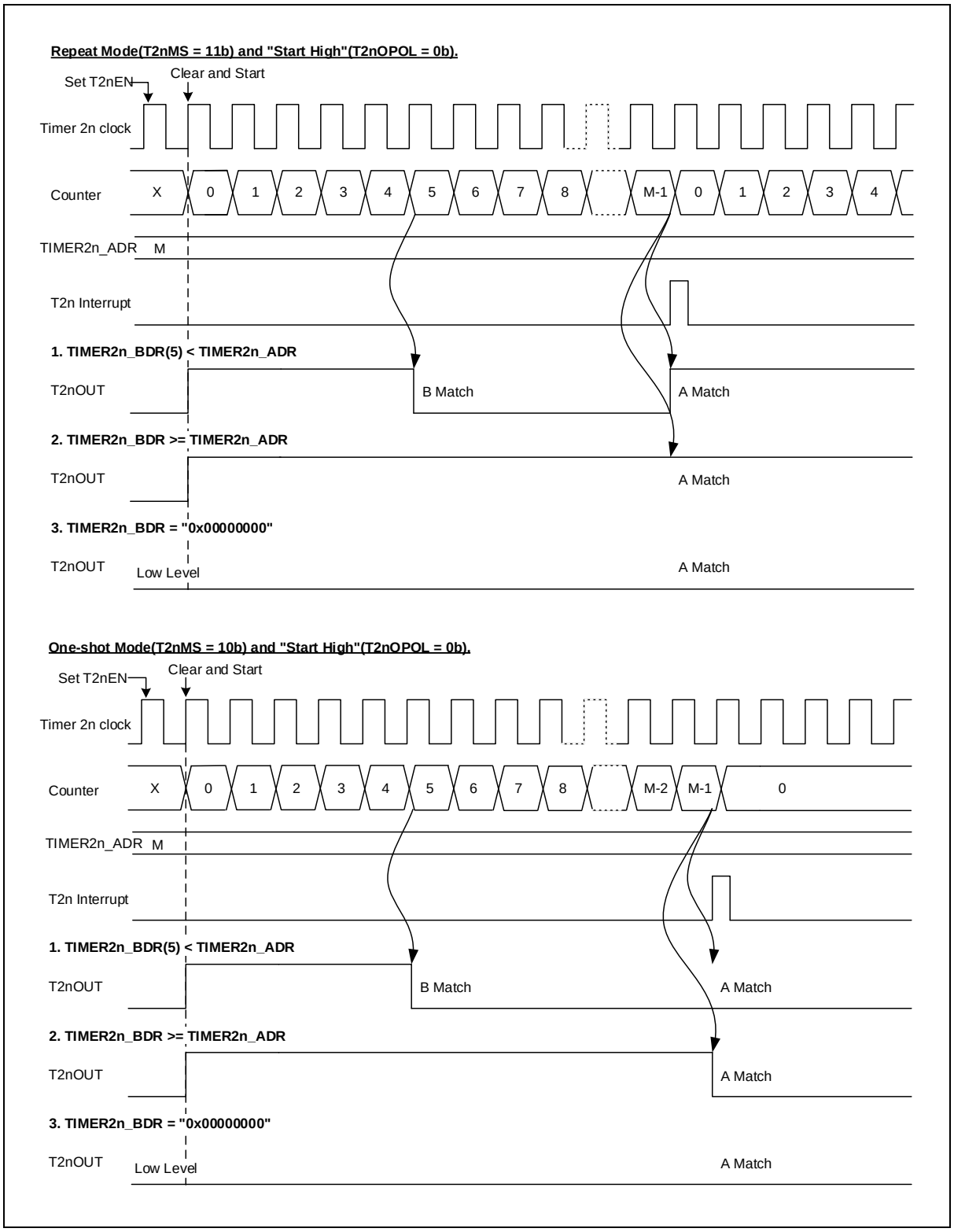


Figure 87. PPG Mode Timing Example of TIMER 2n (n = 0 or 1)

Figure 88 shows the PPG mode operation of timer/counter 2n. Refer to Figure 13 for internal timer clock source and Table 25 for Timer2n Output pin.

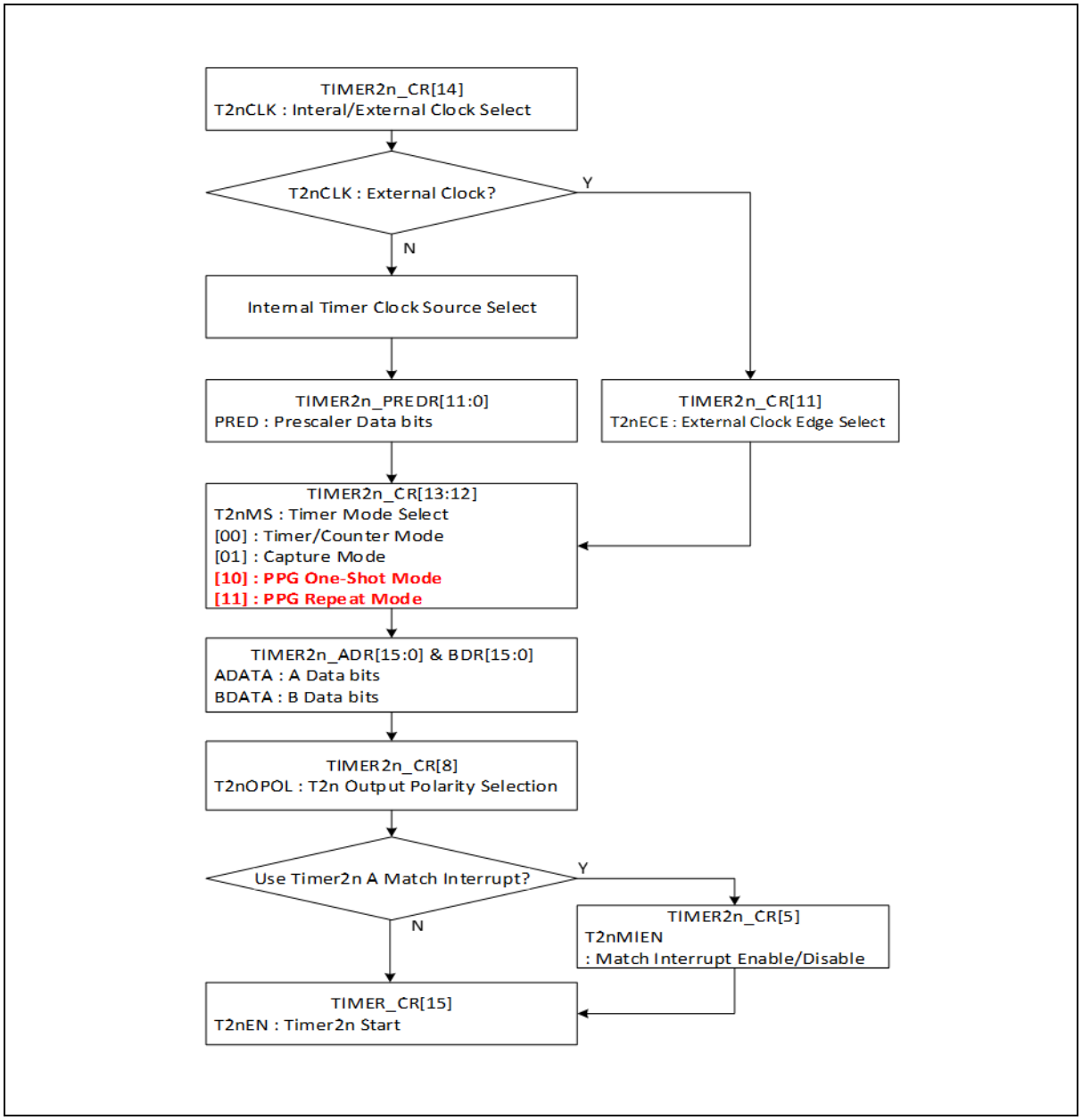


Figure 88. PPG Mode Operation Sequence of TIMER 2n (n = 0 or 1)

13. Timer counter 30

Timer counter 30 of A31G24x series consist of a multiplexer, a comparator, 16-bit sized timer data registers A/B/C, and many other registers used for its operation. Main features are listed in the followings:

- 16-bit up/down-counter
- Periodic timer mode
- Back-to-Back mode
- Capture mode
- 12-bit prescaler

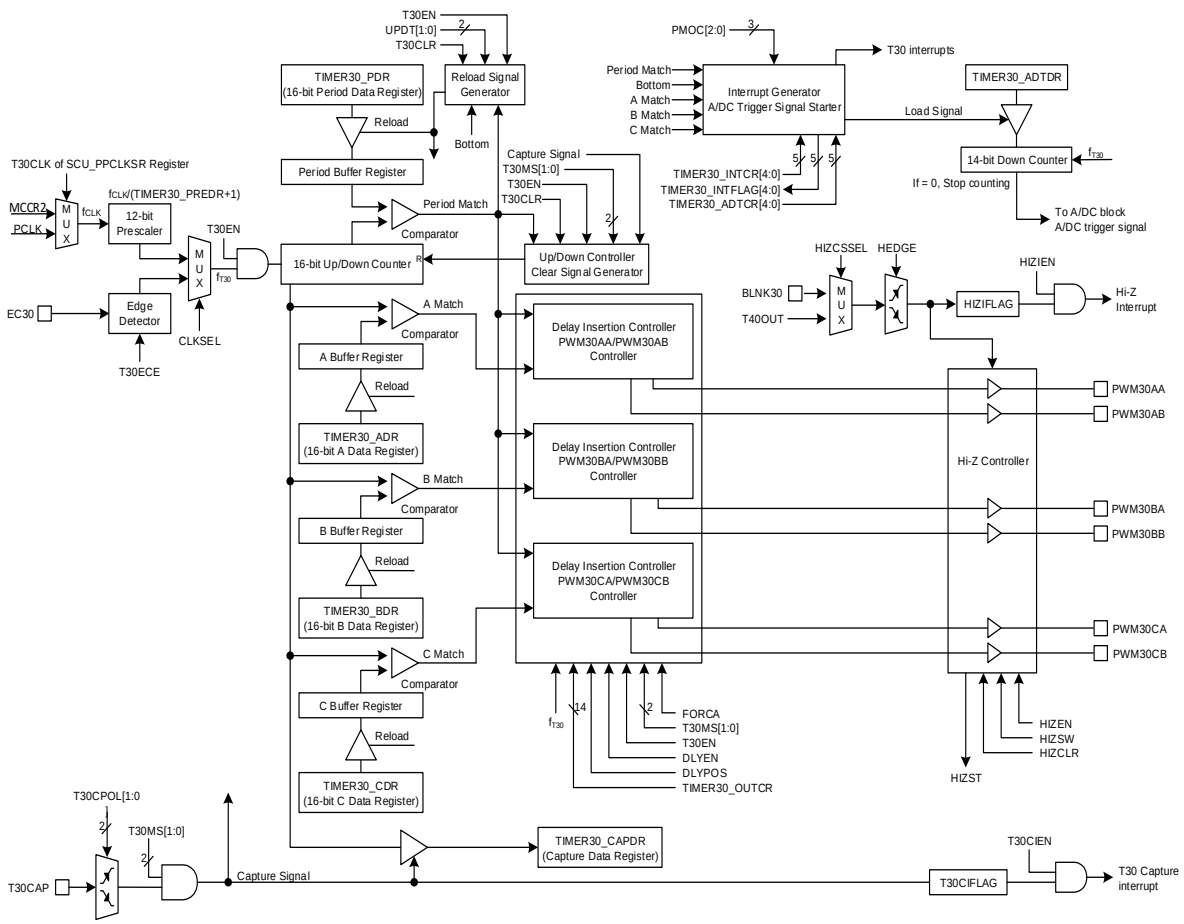
Table 54 introduces pins assigned for the timer counter 30.

Table 54. Pin Assignment of Timer Counter 30: External Pins

Pin name	Type	Description
EC30	I	External clock input
T30CAP	I	Capture input
BLNK30	I	External sync signal input
T40OUT	I	Internal sync signal Input (Timer40 Out)
PWM30AA	O	PWM output
PWM30AB	O	PWM output
PWM30BA	O	PWM output
PWM30BB	O	PWM output
PWM30CA	O	PWM output
PWM30CB	O	PWM output

13.1 Timer counter 30 block diagram

In this section, timer counter 30 is introduced in a block diagram.



NOTES:

1. The Period/A/B/C match interrupts except for Bottom can occur in the interval mode.
2. The Period match interrupt only can occur in the capture mode.
3. The 16-bit up/down counter in the capture mode is cleared to "0x0000" after the counter value is loaded to the TIMER30_CAPDR register at valid edge.
4. The HIZSW (high-impedance output software setting) bit can be used for debugging mode. That is, All T30PWM outputs can be high-impedance state with setting the HIZSW bit to '1' by the UI of debugger.
5. After the T30EN bit is set, the counting of TIMER30_CNT may be delayed 2 clocks or more.

Figure 89. Timer Counter 30 Block Diagram

13.2 Registers

Base address of 3-phase PWM timer 30 is introduced in the followings:

Table 55. Base Address of Timer Counter 30

Name	Base address
TIMER30	0x4000_2400

Table 56. Timer Counter 30 Register Map

Name	Offset	Type	Description	Reset value	Reference
TIMER30_CR	0x0000	RW	Timer/Counter 30 Control Register	0x0000_0000	13.2.1
TIMER30_PDR	0x0004	RW	Timer/Counter 30 Period Data Register	0x0000_FFFF	13.2.2
TIMER30_ADR	0x0008	RW	Timer/Counter 30 A Data Register	0x0000_FFFF	13.2.3
TIMER30_BDR	0x000C	RW	Timer/Counter 30 B Data Register	0x0000_FFFF	13.2.4
TIMER30_CDR	0x0010	RW	Timer/Counter 30 C Data Register	0x0000_FFFF	13.2.5
TIMER30_CAPDR	0x0014	RO	Timer/Counter 30 Capture Data Register	0x0000_0000	13.2.6
TIMER30_PREDR	0x0018	RW	Timer/Counter 30 Prescaler Data Register	0x0000_0FFF	13.2.7
TIMER30_CNT	0x001C	RO	Timer/Counter 30 Counter Register	0x0000_0000	13.2.8
TIMER30_OUTCR	0x0020	RW	Timer/Counter 30 Output Control Register	0x0000_0000	13.2.9
TIMER30_DLY	0x0024	RW	Timer/Counter 30 PWM Output Delay Data Register	0x0000_0000	13.2.10
TIMER30_INTCR	0x0028	RW	Timer/Counter 30 Interrupt Control Register	0x0000_0000	13.2.11
TIMER30_INTFLAG	0x002C	RW	Timer/Counter 30 Interrupt Flag Register	0x0000_0000	13.2.12
TIMER30_HIZCR	0x0030	RW	Timer/Counter 30 High-Impedance Control Register	0x0000_0000	13.2.13

Table 56. Timer Counter 30 Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
TIMER30_ADTCR	0x0034	RW	Timer/Counter 30 A/DC Trigger Control Register	0x0000_0000	13.2.14
TIMER30_ADTDR	0x0038	RW	Timer/Counter 30 A/DC Trigger Generator Data Register	0x0000_0000	13.2.15

13.2.1 TIMER30_CR: Timer/counter 30 control register

Timer module should be configured properly before running. When target purpose is defined, the timer can be configured in this register. After configuring this register, you can start or stop the timer function by TIMER30_CR register.

TIMER30_CR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CR=0x4000_2400

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		T30EN	T30CLK	T30MS		T30ECE	FORCA	DLYEN	DLYPOS	T30CPOL		UPDT		PMOC			T30CLR
-		0	0	00		0	0	0	0	00		00		000			0
-		RW	RW	RW		RW	RW	RW	RW	RW		RW		RW			RW

15	T30EN	Timer 30 Operation Enable bit.	
		0	Disable timer 30 operation.
		1	Enable timer 30 operation. (Counter clear and start)
14	T30CLK	Timer 30 Clock Selection bit.	
		0	Select an internal prescaler clock.
		1	Select an external clock.
NOTES: This bit should be changed during T30EN bit is "0b". If you select an internal prescaler clock, you should set T30CLK bit in the SCU_PPCLKSR register first.			
13 12	T30MS	Timer 30 Operation Mode Selection bits.	
		00	Interval mode. (All match interrupts can occur)
		01	Capture mode. (The Period-match interrupt can occur)
		10	Back-to-back mode. (All interrupts can occur)
		11	Not used.
		NOTE: This bit should be changed during T30EN bit is "0b".	
11	T30ECE	Timer 30 External Clock Edge Selection bit.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.

10	FORCA	Timer 30 Output Mode Selection bit. This bit should be changed when the T30EN is "0b".	
		0	6-Channel mode (The PWM30xA/PWM30xB pins are output according to the TIMER30_xDR registers, respectively)
		1	Force A-Channel mode (The all PWM30xA/PWM30xB pins are output according to the only TIMER30_ADR registers)
9	DLYEN	Delay Time Insertion Enable bit.	
		0	Disable to insert delay time to the PWM30xA/PWM30xB.
		1	Enable to insert delay time to the PWM30xA/PWM30xB.
8	DLYPOS	Delay Time Insertion Position.	
		0	Insert at front of PWM30xA and at back of PWM30xB pins.
		1	Insert at back of PWM30xA and at front of PWM30xB pins.
7 6	T30CPOL	Timer 30 Capture Polarity Selection bits.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both of falling and rising edge.
		11	Reserved
6 4	UPDT	Data Reload Time Selection bits.	
		00	Update data to buffer at the time of writing.
		01	Update data to buffer at period match.
		10	Update data to buffer at bottom.
		11	Not used.
3 1	PMOC	Period Match Interrupt Occurrence Selection.	
		000	Once every period match.
		001	Once every 2 period match.
		010	Once every 3 period match.
		011	Once every 4 period match.
		100	Once every 5 period match.
		101	Once every 6 period match.
		110	Once every 7 period match.
		111	Once every 8 period match.

NOTES:

1. A period match counter is cleared as 0x00 when the T30CLR bit is set.
2. When changing the PMOC value, must clear the period match counter with T30CLR. Otherwise, malfunction may occur.

0	T30CLR	Timer 30 Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer 30 counter and prescaler (Automatically cleared to "0b" after operation)

13.2.2 TIMER30_PDR: Timer/counter 30 period data register

TIMER30_PDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_PDR=0x4000_2404

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		PDATA	
-		0xFFFF	
-		RW	

15 0	PDATA	Timer/Counter 30 Period Data bits. The range is 0x0002 to 0xFFFF.

NOTE: Do not write "0x0000" in the TIMER30_PDR register when PPG mode.

13.2.3 TIMER30_ADR: Timer/counter 30 A data register

TIMER30_ADR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADR=0x4000_2408

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		ADATA	
-		0xFFFF	
-		RW	

15 0	ADATA	Timer/Counter 30 A Data bits. The range is 0x0000 to 0xFFFF.

13.2.4 TIMER30_BDR: Timer/counter 30 B data register

TIMER30_BDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_BDR=0x4000_240C

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		BDATA	
-		0xFFFF	
-		RW	

15 0	BDATA	Timer/Counter 30 B Data bits. The range is 0x0000 to 0xFFFF.

13.2.5 TIMER30_CDR: Timer/counter 30 C data register

TIMER30_CDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CDR=0x4000_2410

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		CDATA	
-		0xFFFF	
-		RW	

15 0	CDATA	Timer/Counter 30 C Data bits. The range is 0x0000 to 0xFFFF.

13.2.6 TIMER30_CAPDR: Timer/counter 30 capture data register

TIMER30_CAPDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CAPDR=0x4000_2414

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		CAPD	
-		0x0000	
-		RO	

15 0	CAPD	Timer/Counter 30 Capture Data bits.

13.2.7 TIMER30_PREDR: Timer/counter 30 prescaler data register

TIMER30_PREDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_PREDR=0x4000_2418

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		PRED	
-		0xFFF	
-		RW	

11 0	PRED	Timer/Counter 30 Prescaler Data bits.

13.2.8 TIMER30_CNT: Timer/counter 30 counter register

TIMER30_CNT is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CNT=0x4000_241C

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		CNT	
-		0x0000	
-		RO	

15 0	CNT	Timer/Counter 30 Counter bits.

13.2.9 TIMER30_OUTCR: Timer/counter 30 output control register

TIMER30_OUTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_OUTCR=0x4000_2420

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
WTIDKY		POLB	POLA	PABOE	PBBOE	PCBOE	PAAOE	PBAOE	PCAOE	Reserved	LVLAB	LVLBB	LVLCB	Reserved	LVLAA	LVLBA	LVLCA
0x0000		0	0	0	0	0	0	0	0	-	0	0	0	-	0	0	0
WO		RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	-	RW	RW	RW

31 16	WTIDKY	Write Identification Key. On writes, write 0xE06C to these bits, otherwise the write is ignored.	
15	POLB	PWM30xB Output Polarity Selection bit. (x : A, B and C)	
		0	Low level start. (The PWM30xB pins are started with low level after counting)
		1	High level start. (The PWM30xB pins are started with high level after counting)
14	POLA	PWM30xA Output Polarity Selection bit. (x : A, B and C)	
		0	Low level start. (The PWM30xA pins are started with low level after counting)
		1	High level start. (The PWM30xA pins are started with high level after counting)
13	PABOE	PWM30AB Output Enable bit.	
		0	Disable output.
		1	Enable output.
12	PBBOE	PWM30BB Output Enable bit.	
		0	Disable output.
		1	Enable output.
11	PCBOE	PWM30CB Output Enable bit.	
		0	Disable output.
		1	Enable output.

10	PAAOE	PWM30AA Output Enable bit.	
		0	Disable output.
		1	Enable output.
9	PBAOE	PWM30BA Output Enable bit.	
		0	Disable output.
		1	Enable output.
8	PCAOE	PWM30CA Output Enable bit.	
		0	Disable output.
		1	Enable output.
6	LVLAB	Configure PWM30AB output When Disable.	
		0	Low level
		1	High level
5	LVLBB	Configure PWM30BB output When Disable.	
		0	Low level
		1	High level
4	LVLCB	Configure PWM30CB output When Disable.	
		0	Low level
		1	High level
2	LVLAA	Configure PWM30AA output When Disable.	
		0	Low level
		1	High level
1	LVLBA	Configure PWM30BA output When Disable.	
		0	Low level
		1	High level
0	LVLCA	Configure PWM30CA output When Disable.	
		0	Low level
		1	High level

13.2.10 TIMER30_DLY: Timer/counter 30 PWM output delay data register

TIMER30_DLY is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_DLY=0x4000_2424

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			DLY
-			0x000
-			RW

9 0	DLY	Timer/Counter 30 PWM Delay Data bits. Delay time: (DLY[9:0]+1)÷fT30

13.2.11 TIMER30_INTCR: Timer/counter 30 interrupt control register

TIMER30_INTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_INTCR=0x4000_2428

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				HIZIEN	T30CIEN	T30BTIEN	T30PMIEN	T30AMIEN	T30BMIEN	T30CMIEN
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	HIZIEN	Timer 30 Output High-Impedance Interrupt Enable bit.	
		0	Disable timer 30 output high-impedance interrupt.
		1	Enable timer 30 output high-impedance interrupt.
5	T30CIEN	Timer 30 Capture Interrupt Enable bit.	
		0	Disable timer 30 capture interrupt.
		1	Enable timer 30 capture interrupt.
4	T30BTIEN	Timer 30 Bottom Interrupt Enable bit.	
		0	Disable timer 30 period interrupt.
		1	Enable timer 30 period interrupt.
3	T30PMIEN	Timer 30 Period Match Interrupt Enable bit.	
		0	Disable timer 30 period interrupt.
		1	Enable timer 30 period interrupt.
2	T30AMIEN	Timer 30 A-ch Match Interrupt Enable bit.	
		0	Disable timer 30 A-ch match interrupt.
		1	Enable timer 30 A-ch match interrupt.
1	T30BMIEN	Timer 30 B-ch Match Interrupt Enable bit.	
		0	Disable timer 30 B-ch match interrupt.
		1	Enable timer 30 B-ch match interrupt.
0	T30CMIEN	Timer 30 C-ch Match Interrupt Enable bit.	
		0	Disable timer 30 C-ch match interrupt.
		1	Enable timer 30 C-ch match interrupt.

13.2.12 TIMER30_INTFLAG: Timer/counter 30 interrupt flag register

TIMER30_INTFLAG is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_INTFLAG=0x4000_242C

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				HIZIFLAG	T30CIFLAG	T30BTIFLAG	T30PMIFLAG	T30AMIFLAG	T30BMIFLAG	T30CMIFLAG
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	HIZIFLAG	Timer 30 Output High-Impedance Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
5	T30CIFLAG	Timer 30 Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
4	T30BTIFLAG	Timer 30 Bottom Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3	T30PMIFLAG	Timer 30 Period Match Flag Enable bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
2	T30AMIFLAG	Timer 30 A-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
1	T30BMIFLAG	Timer 30 B-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
0	T30CMIFLAG	Timer 30 C-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.

13.2.13 TIMER30_HIZCR: Timer/counter 30 high-impedance control register

TIMER30_HIZCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_HIZCR=0x4000_2430

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			HIZCSSEL	HIZEN	Reserved	HIZSW	Reserved	HEDGE	HIZSTA	HIZCLR
-			0	0	-	0	-	0	0	0
-			RW	RW	-	RW	-	RW	RO	RW

8	HIZCSSEL	High-Impedance output control signal selection	
		0	BLNK30
		1	T40OUT
7	HIZEN	PWM30xA/PWM30xB Output High-Impedance Enable bit.	
		0	Disable to control the output high-impedance.
		1	Enable to control the output high-impedance.
4	HIZSW	High-Impedance Output Software Setting.	
		0	No effect.
		1	PWM30xA/PWM30xB pins go into high impedance. (Automatically cleared to "0b" after operation)
2	HEDGE	High-Impedance Edge Selection.	
		0	Falling edge of the BLNK30 pin.
		1	Rising edge of the BLNK30 pin.
1	HIZSTA	High-Impedance Status.	
		0	Indicates that the pins are not under a Hi-Z state.
		1	Indicates that the pins are under a Hi-Z state.
0	HIZCLR	High-Impedance Output Clear bit.	
		0	No effect.
		1	Clear high-impedance output. (The PWM30xA/PWM30xB pins are back to output and this bit is automatically cleared to "0b" after operation)
NOTE: Where x = A, B, and C.			

13.2.14 TIMER30_ADTCR: Timer/counter 30 A/DC trigger control register

TIMER30_ADTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADTCR=0x4000_2434

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				T30BTTG	T30PMTG	T30AMTG	T30BMTG	T30CMTG
-				0	0	0	0	0
-				RW	RW	RW	RW	RW

4	T30BTTG	Select Timer 30 Bottom for A/DC Trigger Signal Generator.	
		0	Disable A/DC trigger signal generator by bottom.
		1	Enable A/DC trigger signal generator by bottom.
3	T30PMTG	Select Timer 30 Period Match for A/DC Trigger Signal Generator.	
		0	Disable A/DC trigger signal generator by period match.
		1	Enable A/DC trigger signal generator by period match.
2	T30AMTG	Select Timer 30 A-ch Match for A/DC Trigger Signal Generator.	
		0	Disable A/DC trigger signal generator by A-ch match.
		1	Enable A/DC trigger signal generator by A-ch match.
1	T30BMTG	Select Timer 30 B-ch Match for A/DC Trigger Signal Generator.	
		0	Disable A/DC trigger signal generator by B-ch match.
		1	Enable A/DC trigger signal generator by B-ch match.
0	T30CMTG	Select Timer 30 C-ch Match for A/DC Trigger Signal Generator.	
		0	Disable A/DC trigger signal generator by C-ch match.
		1	Enable A/DC trigger signal generator by C-ch match.

NOTES:

1. A trigger signal generation is not related with the PMOC[2:0] bits of TIMER30_CR register.
2. If several source for trigger is selected, a signal can be lost in case of the trigger generation counter is reloaded by another signal.

13.2.15 TIMER30_ADTDR: Timer/counter 30 A/DC trigger generator data register

TIMER30_ADTDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADTDR=0x4000_2438

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		ADTDATA	
-		0x0000	
-		RW	

13 0	CNT	Timer/Counter 30 A/DC Trigger Generation Data bits.

NOTES:

1. ADTDR of Timer30 uses the adcnt timer as tclk (timer30 clock).
2. The adcnt timer counter is a 14-bit down counter.
3. It count down from the value written in ADTDR to 0. (Delay role).

13.3 Functional description

13.3.1 Timer counter 30

The timer/counter 30 can be clocked by an internal or an external clock source (EC30). The clock source is selected by a clock selection logic which is controlled by the clock selection bits (T30CLK).

- TIMER 30 clock source: PCLK/(TIMER30_PREDR +1), EC30

In capture mode, by T30CAP, data is captured into input capture data register (TIMER30_CAPDR).

The PWM wave form to PWM30AA, PWM30AB, PWM30BA, PWM30BB, PWM30CA, PWM3CB Port (6-channel).

Table 57. Timer 30 Operating Modes

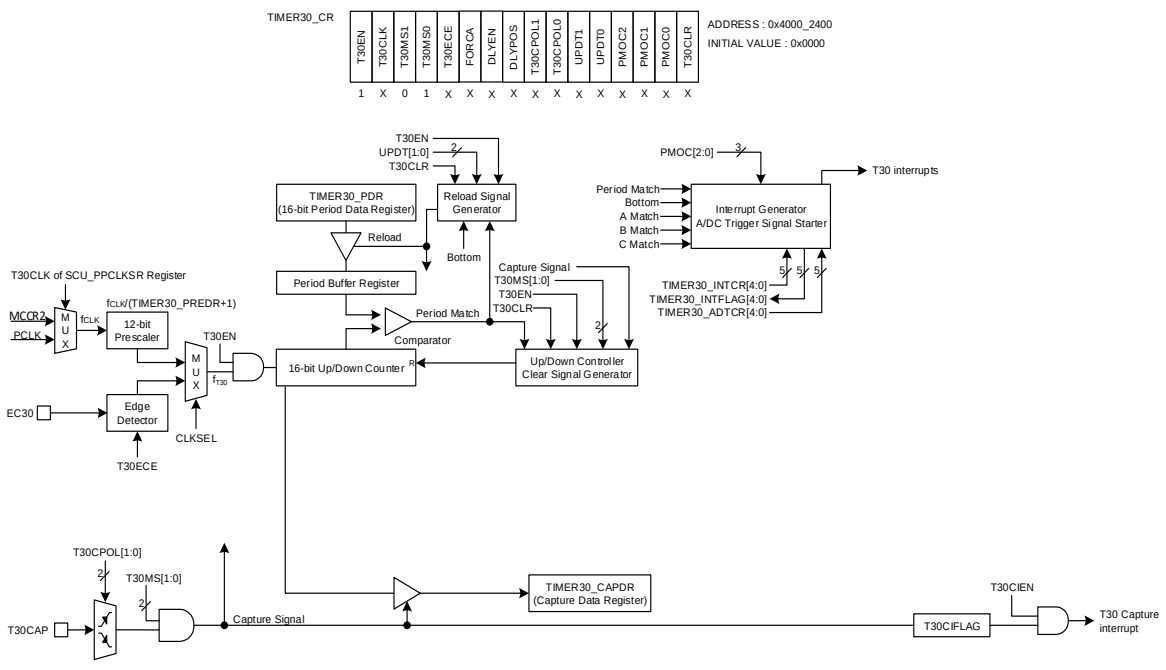
T30EN	Alternative Mode	T30MS[1:0]	TIMER30_PREDR	Timer 30 MODE
1	AF3	00	0xXXX	16-bit Interval Mode
1	AF4	01	0xXXX	16-bit Capture Mode
1	AF3	10	0xXXX	16-bit back-to-back Mode

13.3.2 Timer 30 capture mode

16-bit timer 30 capture mode is set by configuring T30MS[1:0] as '01'. An internal clock input or an external clock input can be used as a clock source. Basically, the 16-bit timer 30 capture mode has the same function as the 16-bit interval mode has. Interrupts occur when value of TIMER30's 16-bit up/down counter equals to the one of TIMER30_PDR. The 16-bit up/down counter values are automatically cleared by a match signal. It can be cleared by software (T30CLR) too.

The 16-bit timer 30's interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER30_CAPDR.

Figure 90 shows 16-bit capture mode of the timer 30.



NOTES:

1. The Period match only can occur in the capture mode.
2. The 16-bit up/down counter in the capture mode is cleared to "0x0000" after the counter value is loaded to the TIMER30_CAPDR register at valid edge.

Figure 90. 16-bit Capture Mode of Timer 30

Figure 91 shows the capture mode operation of 32-bit Timer30. Refer to Figure 13 for internal timer clock source and Table 25 for Timer 30 capture pin.

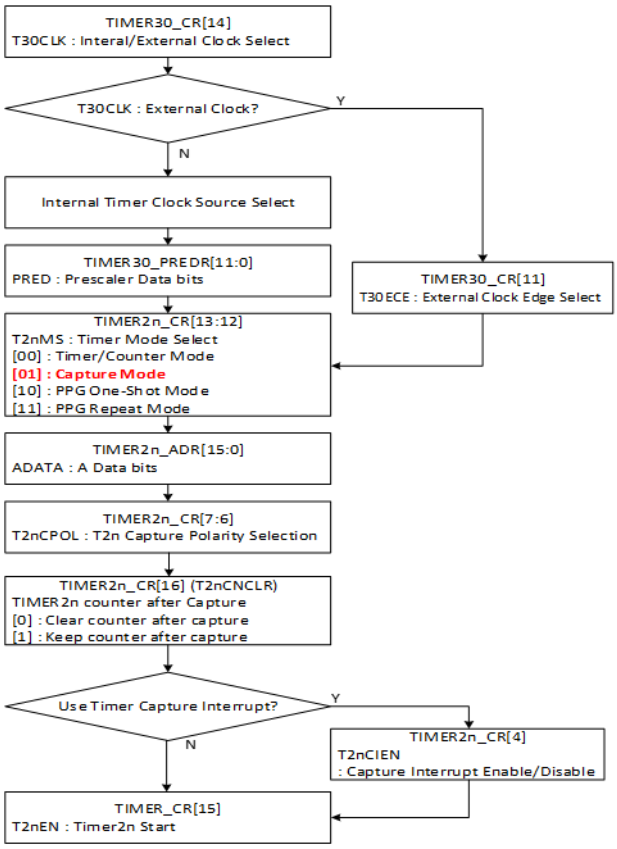


Figure 91. Capture Mode Operation Sequence of TIMER 30

13.3.3 Timer 30 interval mode

Timer 30 interval mode is set by configuring T30MS[1:0] as '00'. The timer 30 has a counter and data registers. The 16-bit up/down counter is increased by an internal or an external clock input. The timer 30 can use the input clock with 12-bit prescaler division rates (TIMER30_PREDR[11:0]). When the value of TIMER30 16-bit up/down counter and the value of TIMER30_PDR are identical in timer 30, a match signal is generated and the period match interrupt of timer 30 is occurred. The period match interrupt can be occurred which once every 1, 2, 3, 4, 5, 6, 7, or 8 period match (PMOC[2:0]). The 16-bit up/down counter value is automatically cleared by match signal. It can be cleared by software (T30CLR) too.

The timer 30 Interval mode can be operated for BLDC motor control. It has 6-channel pins output up to 16-bit resolution PWM output. When the value of 16-bit up/down counter and TIMER30_PDR are identical in timer 30, a period match signal is generated and the period match interrupt of timer 30 is occurred.

The timer 30 A, B, and C match signals are generated and the A, B, and C match interrupts of timer 30 are occurred, when the 16-bit counter value are identical to the value of TIMER30_xDR. The period and duty of the PWM output is determined by the TIMER30_PDR (PWM period register), and T3xDR (each channel PWM duty register).

- PWM Period = [TIMER30_PDR] X Source Clock
- PWM Duty(A-ch) = [TIMER30_ADR] X Source Clock
- PWM Duty(B-ch) = [TIMER30_BDR] X Source Clock
- PWM Duty(C-ch) = [TIMER30_CDR] X Source Clock

The POLA/POLB bit of TIMER30_OUTCR register decides the polarity of PWM output. If the POLA/POLB bit is set to '1b', the PWM30xA/PWM30xB output is high level start, respectively. And if the POLA/POLB bit is cleared to '0b', the PWM30xA/PWM30xB output is low level start, respectively.

Table 58. PWM Channel Polarity

PxAOE	PxBOE	POLxA	POLxB	PWM3xA Pin put	PWM3xB Pin Output
1	1	0	0	Low level start	Low level start
		0	1	Low level start	High level start
		1	0	High level start	Low level start
		1	1	High level start	High level start

NOTE:

1. Where x = A, B, and C.

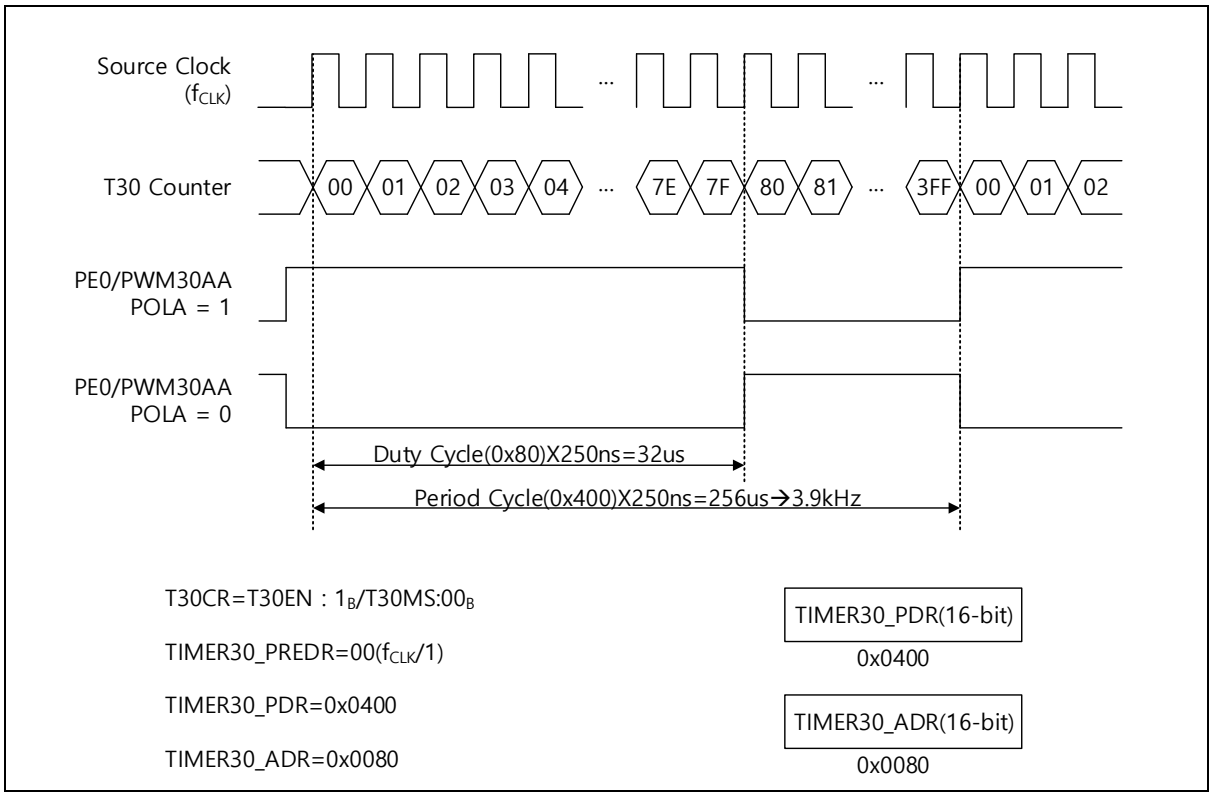
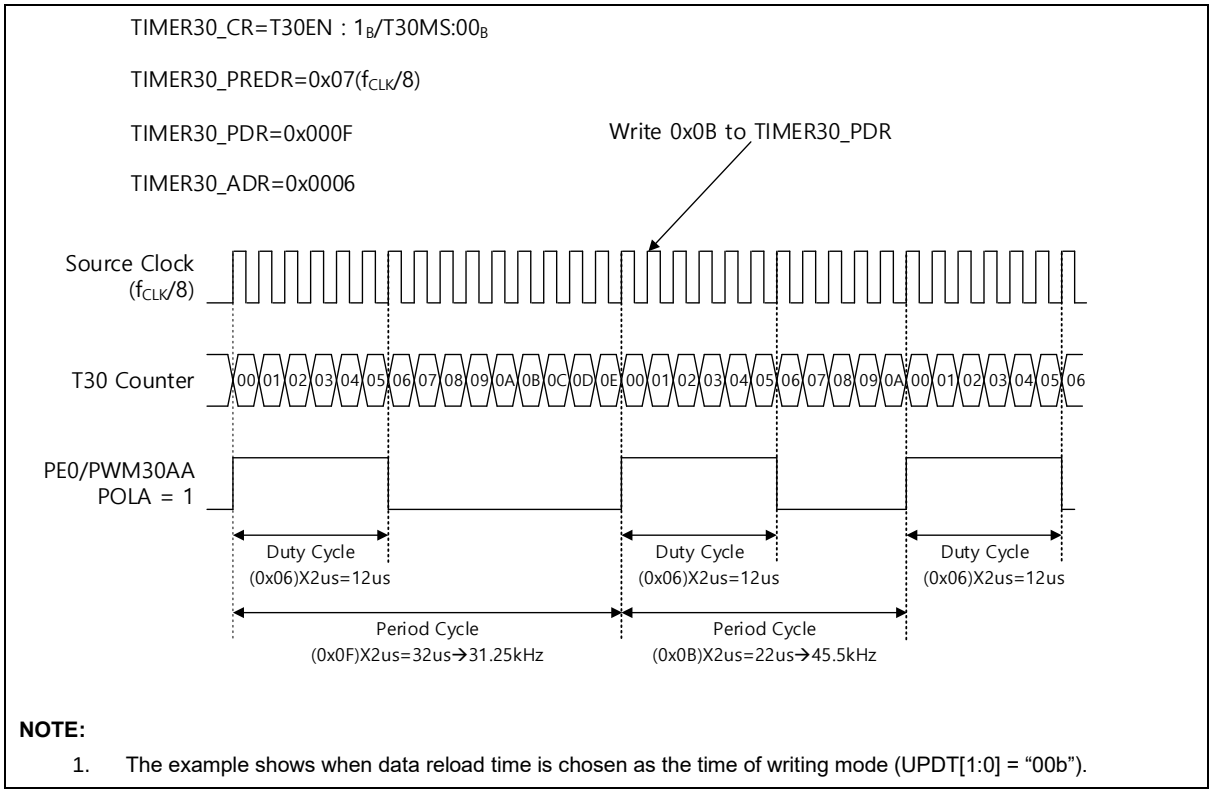


Figure 92. Example of PWM at 4MHZ



NOTE:

1. The example shows when data reload time is chosen as the time of writing mode (UPDT[1:0] = "00b").

Figure 93. Example of Changing the Period in Absolute Duty Cycle at 4MHz

Data reload time selection

The data reload time can choose among "update data to buffer at the time of writing", "update data to buffer at period match", or "update data to buffer at bottom".

PWM output delay

If using the DLYEN bit, DLYPOS bit, and TIMER30_DLY register, it can delay the PWM output. The DLYPOS setting to '0', the delay inserts at front of PWM30xA and at back of PWM30xB pins. The DLYPOS setting to '1', the delay inserts at back of PWM30xA and at front of PWM30xB pins. The settings of DLYEN bit, DLYPOS bit, and TIMER30_DLY register are applied equally to all PWM channels.

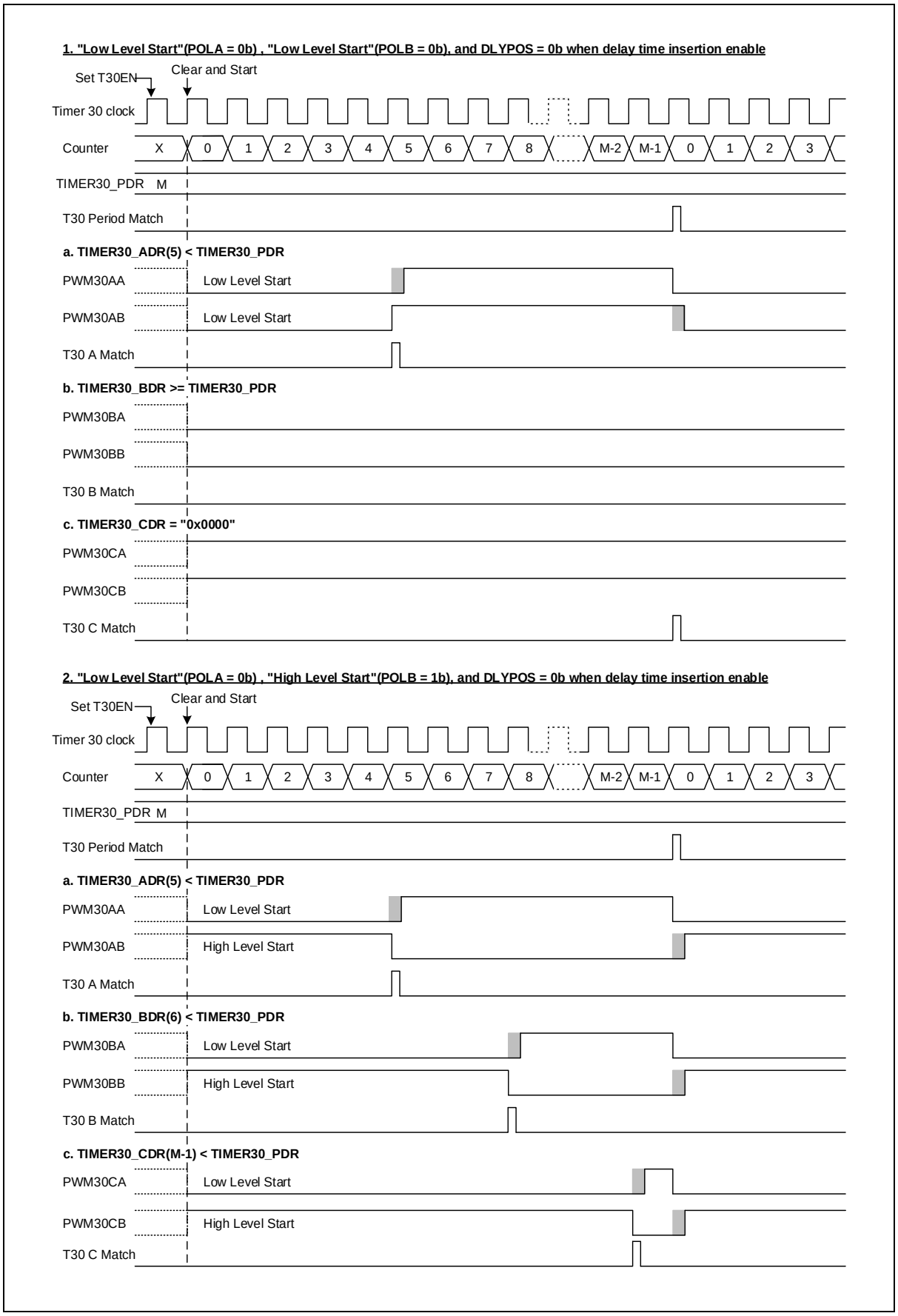


Figure 94. Interval Mode Timing Chart With "DLYPOS = 0"

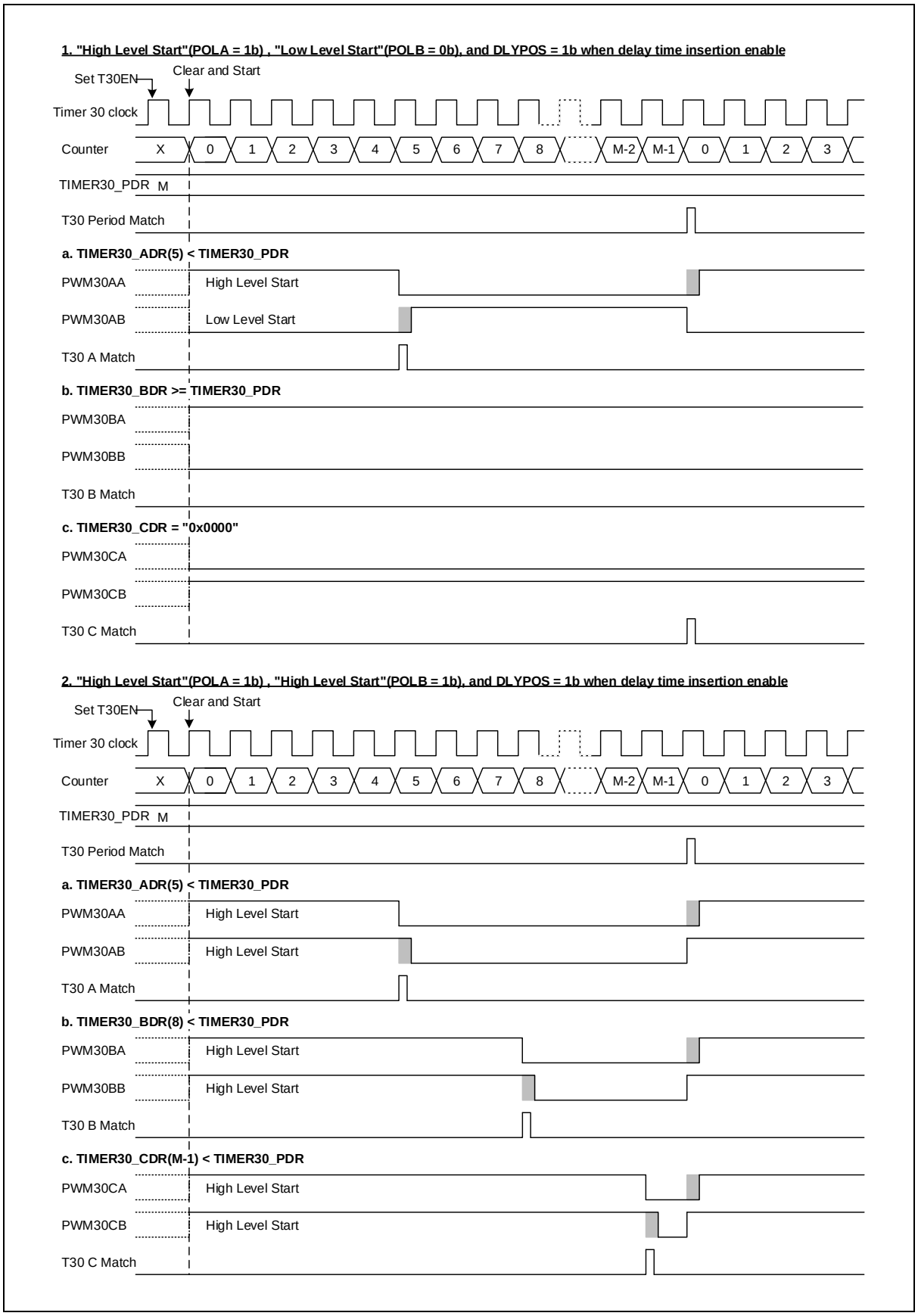


Figure 95. Interval Mode Timing Chart With "DLYPOS = 1

Back-to-back mode

The back-to-back mode is set by T30MS[1:0] as '10'. In the back-to-back mode, the 16-bit up/down counter repeats up/down count. In fact, the effective duty and period becomes twofold of the register set values. If the TIMER30_PDR's data value is set to "0x3210, 16-bit up/down counter will increase until it reaches 0x3210. At this point, a period match signal is generated and the period match interrupt occurs. And then the 16-bit up/down counter will decrease until it reaches 0x0000. At this point, the bottom interrupt occurs. It is repeated in this way.

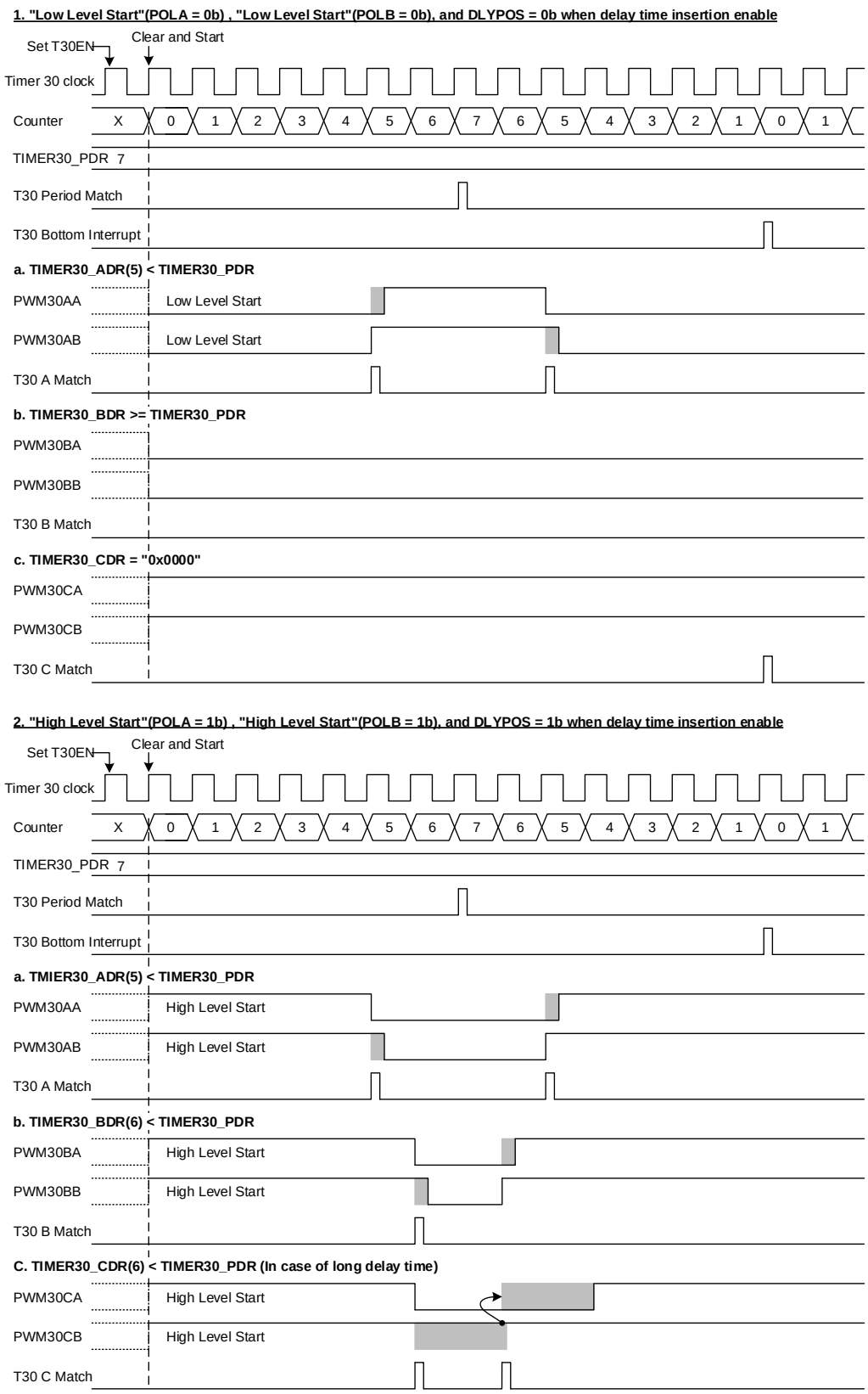


Figure 96. Back-to-Back Mode Timing Chart

Emergency protective function

This protective function is used for emergency stop, when the PWM30xA/PWM30xB output high-impedance enable bit, HIZEN is enabled. When the signal on the external BLNK input pin or internal comparator 3 output goes active (falling or rising edge triggered), the PWM30xA/PWM30xB ports are immediately disabled high-impedance against output and a high-impedance interrupt is occurred. The TIMER30_HIZCR register is used for high-impedance control. The high-impedance source is the external BLNK input pin. The high-impedance edge can be selected by HEDGE bit as falling or rising edge. If the HIZST read value is '1', it indicates that the pins are under a high-impedance state. To return from the high-impedance state, the HIZCLR bit set to '1'. If HIZSW bit is set to '1', PWM30xA/PWM30xB pins go into high impedance by software. It can be used for debugging. (x: A, B and C).

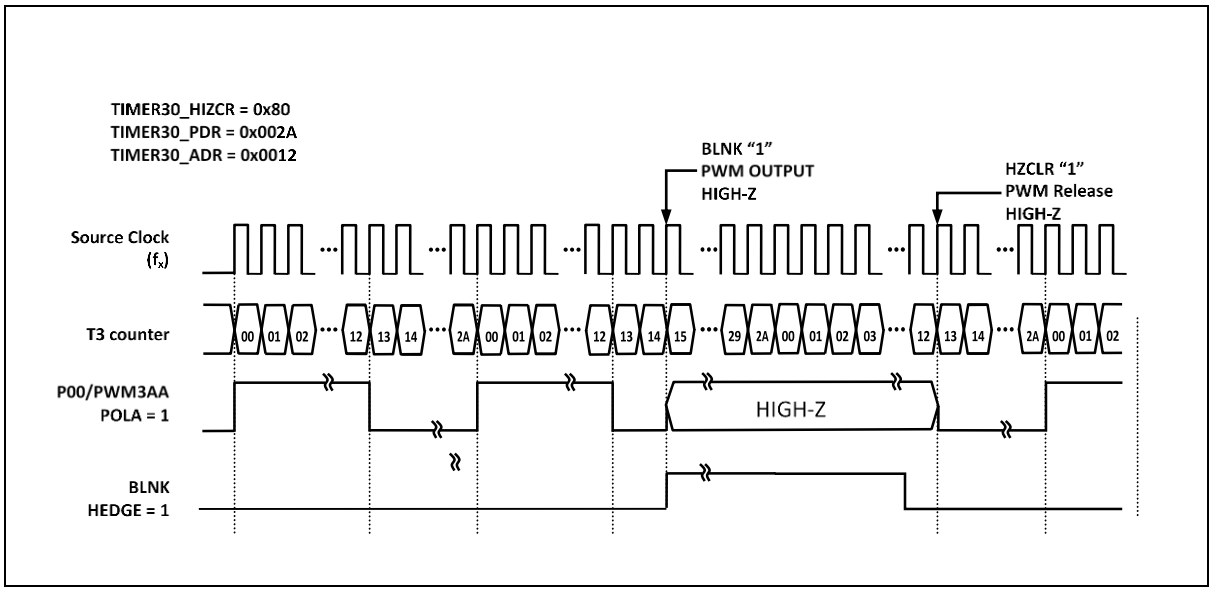


Figure 97. Example of PWM External Synchronization with BLNK Input (x: A, B and C)

Force A-channel mode

If FORCA bit is set to '1', it is possible to enable or disable all PWM output pins through PWM outputs which occur from A-ch duty counter. It is noted that the inversion outputs of A, B, C channel have the same A-ch output waveform.

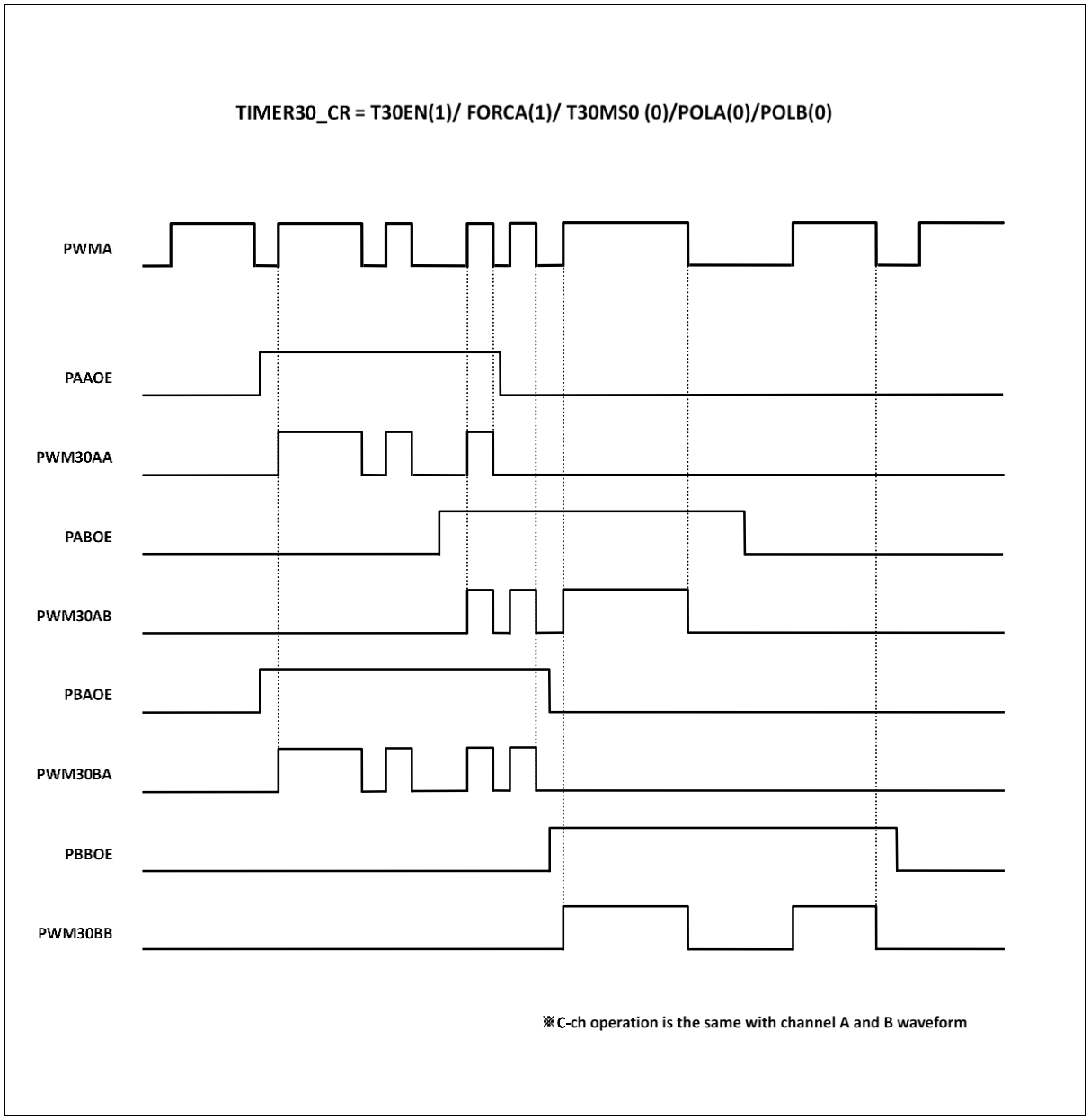


Figure 98. Example of Force A-Channel Mode

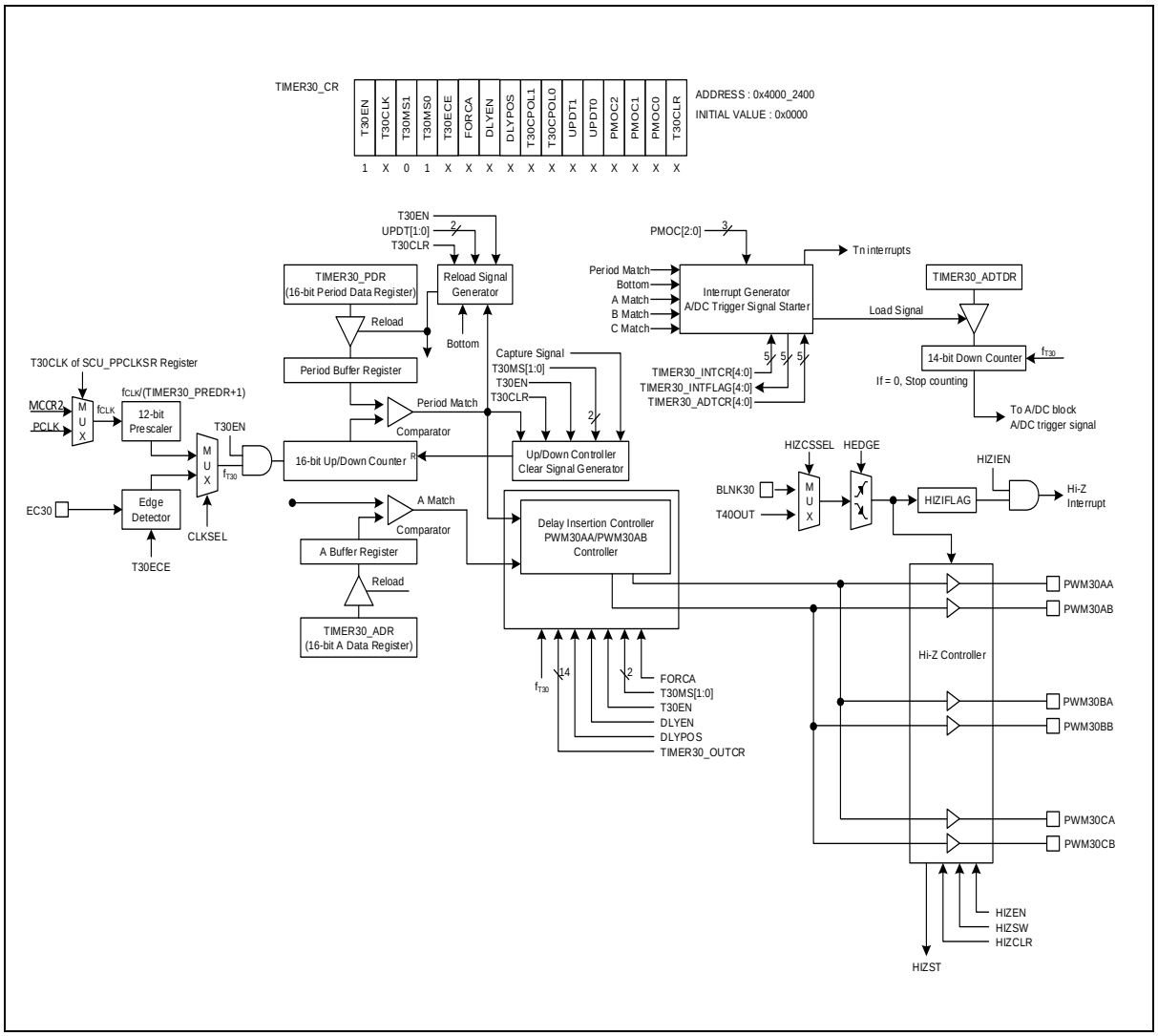


Figure 99. Force A-Channel Mode Block Diagram

6-channel mode

If FORCA bit sets to '0', it is possible to enable or disable PWM output pin and inversion output pin generated through the duty counter of each channel. The inversion output is the reverse phase of the PWM output. A AA/AB output of the A-channel duty register, a BA/BB output of the B-channel duty register, a CA/CB output of the C-channel duty register are controlled respectively.

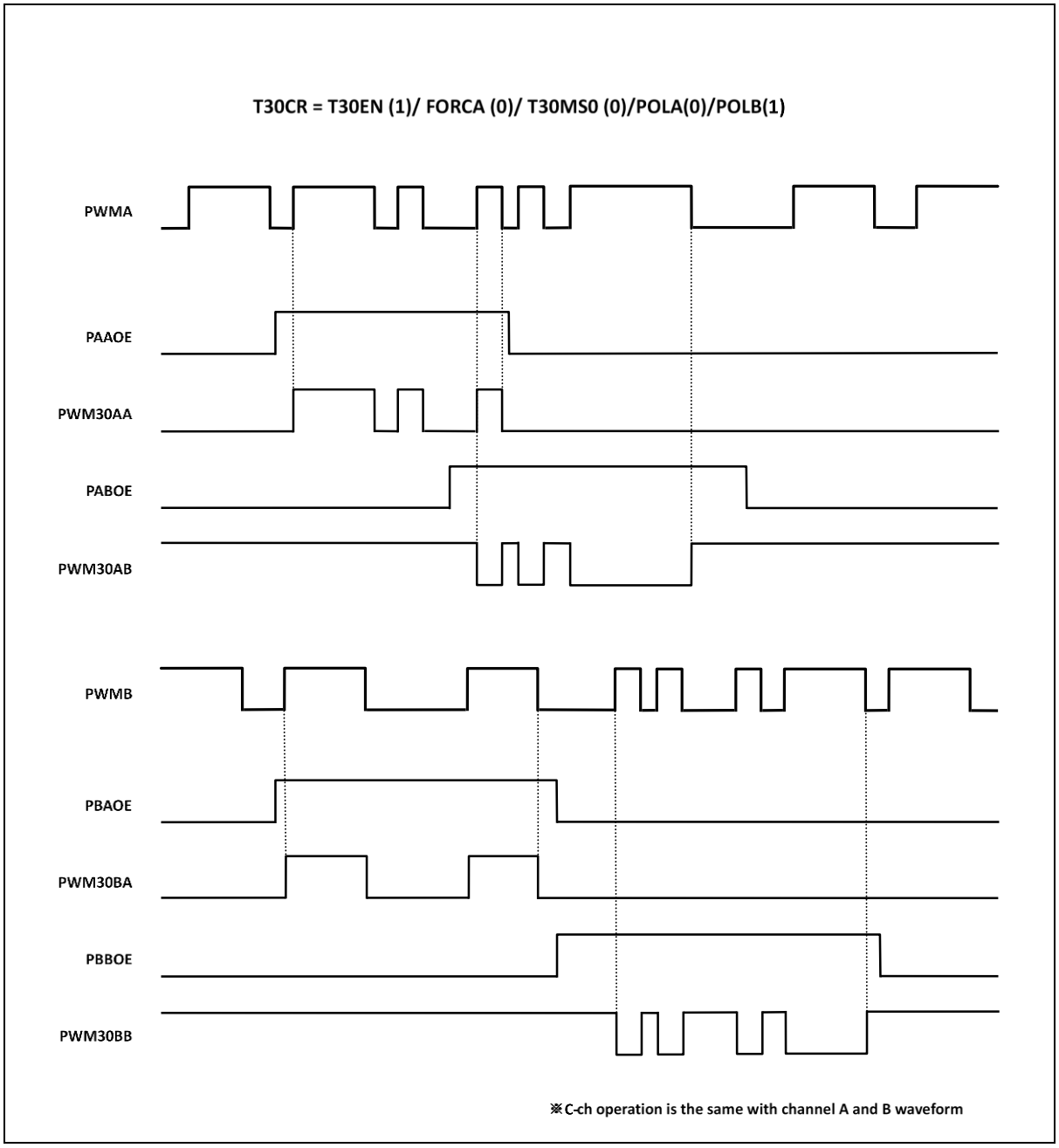


Figure 100. Example of 6-Channel Mode

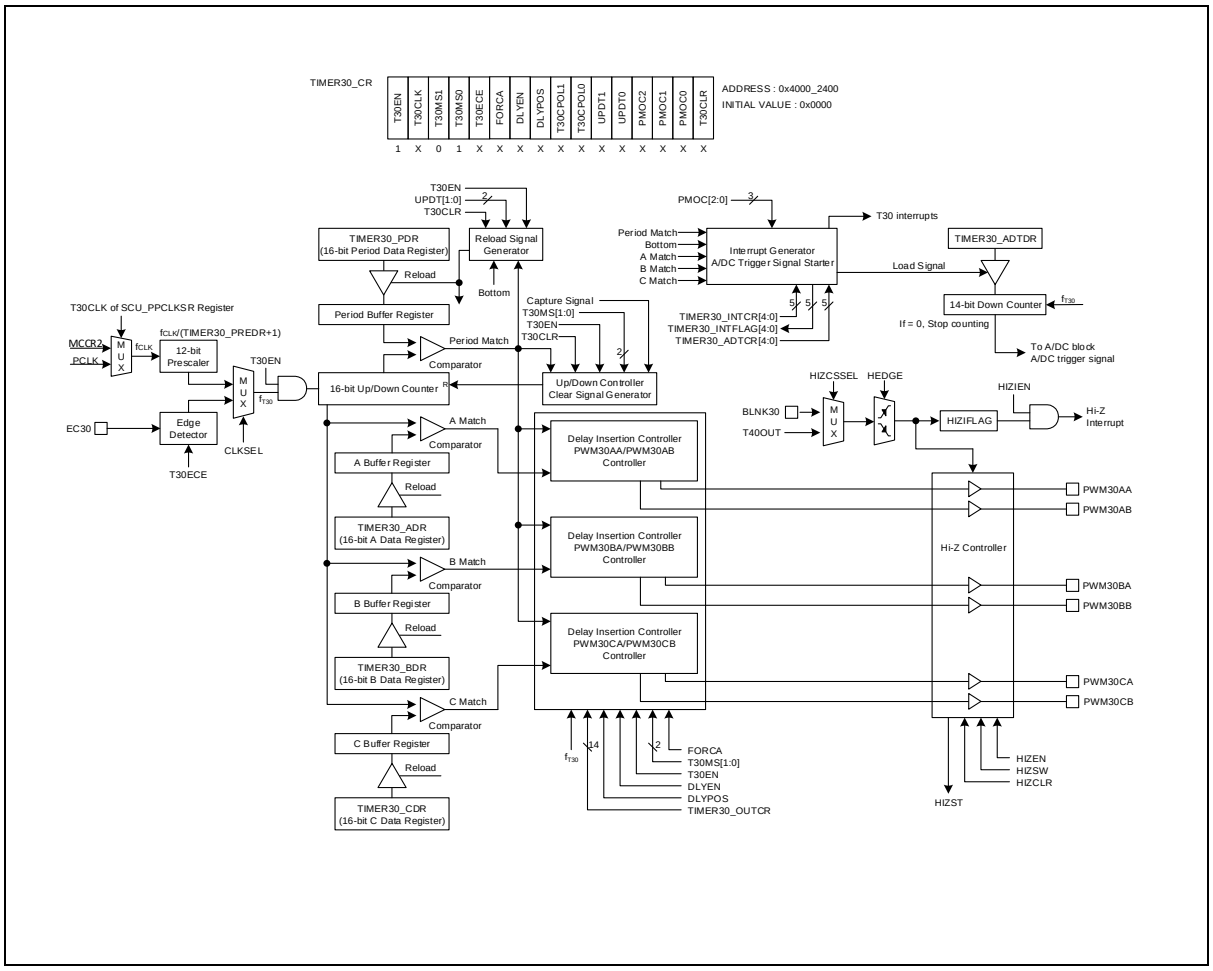


Figure 101. 6-Channel Mode Block Diagram

14. Timer counter 40

Timer counter 40 block comprises a single channel 16-bit general purpose timer. This has an independent 32-bit counter and a dedicated prescaler feeds counting clock. It supports periodic timer, PWM pulse, one-shot timer and capture mode. In addition, optional free-run timer is provided.

Main purpose of the timer counter 40 is a periodical tick timer, and main features are listed in the followings:

- 16-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- 3-Channel XOR Capture mode
- 12-bit prescaler
- Synchronous start and clear function

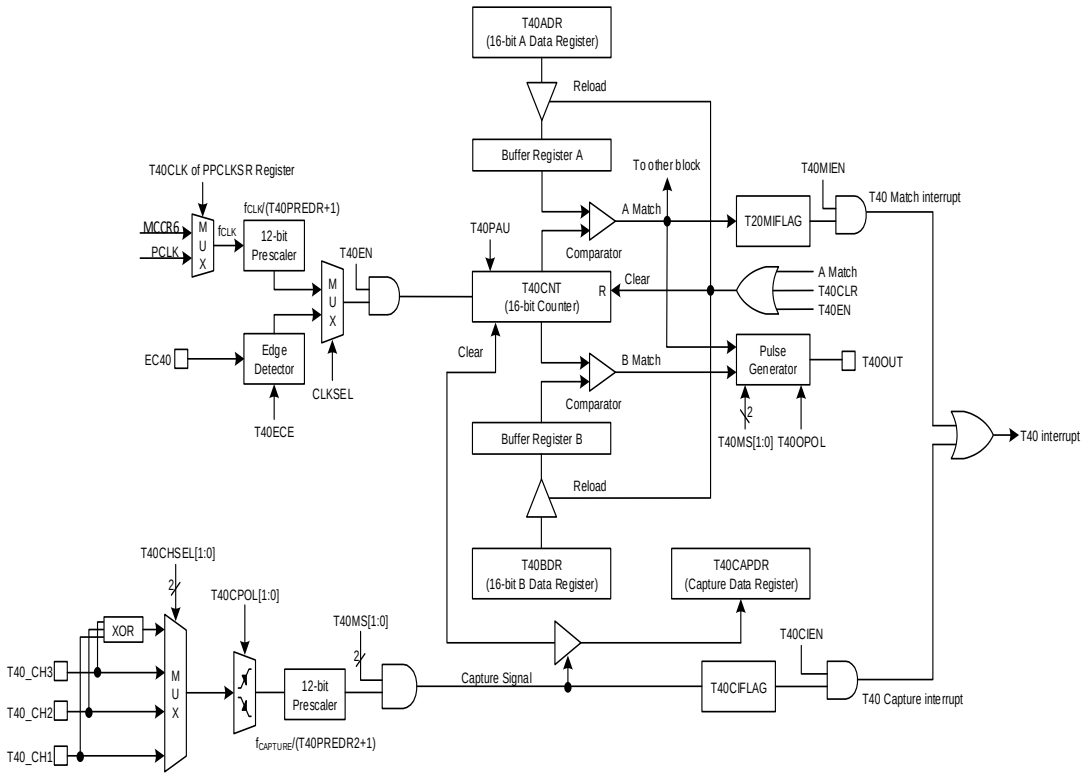
Table 59 introduces pins assigned for the timer counter 40.

Table 59. Pin Assignment of Timer Counter 40: External Pins

Pin name	Type	Description
EC40	I	External clock input
T40_CH1	I	Capture input
T40_CH2	I	Capture input
T40_CH3	I	Capture input
T40OUT	O	Timer/PWM/one-shot output

14.1 Timer counter 40 block diagram

In this section, timer counter 40 is introduced in a block diagram.



NOTES:

1. The T40EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the T40EN bit is set, the counting of TIMER40_CNT may be delayed 2 clocks or more.

Figure 102. Timer Counter 40 Block Diagram

14.2 Registers

Base address of 16-bit timer 40 is introduced in the followings:

Table 60. Base Address of Timer Counter 40

Name	Base address
TIMER40	0x4000_2800

Table 61. Timer Counter 30 Register Map

Name	Offset	Type	Description	Reset value	Reference
TIMER40_CR	0x0000	RW	Timer/Counter 40 Control Register	0x0000_0000	14.2.1
TIMER40_ADR	0x0004	RW	Timer/Counter 40 A Data Register	0xFFFF_FFFF	14.2.2
TIMER40_BDR	0x0008	RW	Timer/Counter 40 B Data Register	0xFFFF_FFFF	14.2.3
TIMER40_CAPDR	0x000C	RO	Timer/Counter 40 Capture Data Register	0x0000_0000	14.2.4
TIMER40_PREDR	0x0010	RW	Timer/Counter 40 Prescaler Data Register	0x0000_0FFF	14.2.5
TIMER40_PREDR 2	0x0014	RW	Timer/Counter 40 Prescaler Data Register 2	0x0000_0FFF	14.2.6
TIMER40_CNT	0x0018	RO	Timer/Counter 40 Counter Register	0x0000_0000	14.2.7

14.2.1 TIMER40_CR: Timer/counter 40 control register

Timer module should be configured properly before running. When target purpose is defined, the timer can be configured in this register. After configuring this register, you can start or stop the timer function by TIMER40_CR register.

TIMER40_CR is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_CR=0x4000_2800

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		T40EN	T40CLK	T40MS	T40ECE	T40CAPCHSEL	T40OPOL	T40CPOL	T40MIEN	T40CIEN	T40MIFLAG	T40CIFLAG	T40PAU	T40CLR
-		0	0	00	0	00	0	00	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	T40EN	Timer 40 Operation Enable bit.	
		0	Disable timer 40 operation.
		1	Enable timer 40 operation. (Counter clear and start)
14	T40CLK	Timer 40 Clock Selection bit.	
		0	Select an internal prescaler clock.
		1	Select an external clock.

NOTES:

1. This bit should be changed during T40EN bit is "0b".
2. If you select an internal prescaler clock, you should set T40CLK bit in the SCU_PPCLKSR register first.

13 12	T40MS	Timer 40 Operation Mode Selection bits.	
		00	Timer/Counter mode. (T40O: Toggle at A-match)
		01	Capture mode. (The A-match interrupt can occur)
		10	PPG one-shot mode. (T40O: Programmable pulse output)
		11	PPG repeat mode. (T40O: Programmable pulse output)
		Note) This bit should be changed during T40EN bit is "0b".	
11	T40ECE	Timer 40 External Clock Edge Selection bit.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.
10 9	T40CAPCHSEL	Timer 40 Capture Channel Selection bits.	
		00	T40_CH1 Select
		01	T40_CH2 Select
		10	T40_CH3 Select
		11	T40 CH1 XOR CH2 XOR CH3 Select
		Note) This bit should be changed during T40EN bit is "0b".	
8	T40OPOL	T40O Polarity Selection bit.	
		0	Start high. (T40O is low level at disable)
		1	Start low. (T40O is high level at disable)

7 6	T40CPOL	Timer 40 Capture Polarity Selection bits.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both of falling and rising edge.
		11	Reserved.
5	T40MIEN	Timer 40 Match Interrupt Enable bit.	
		0	Disable timer 40 match interrupt.
		1	Enable timer 20 match interrupt.
4	T40CIEN	Timer 40 Capture Interrupt Enable bit.	
		0	Disable timer 40 capture interrupt.
		1	Enable timer 20 capture interrupt.
3	T40MIFLAG	Timer 40 Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
2	T40CIFLAG	Timer 40 Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
1	T40PAU	Timer 40 Counter Temporary Pause Control bit.	
		0	Continue counting.
		1	Temporary pause.
0	T40CLR	Timer 40 Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer40 counter and clock prescaler and capture clock presclaer. (Automatically cleared to "0b" after operation)

NOTES:

1. When using ECn (external clock) as a timer clock source in capture mode, fclk (CCR6 or PCLK) must be activated for the capture mode. This is because the fclk is used as a capture clock which is controlling the capture operation.
2. If the internal clock is used as a timer clock source, the fclk is active and the capture mode runs normally.

14.2.2 TIMER40_ADR: Timer/counter 40 A data register

TIMER40_ADR is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_ADR=0x4000_2804

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		ADATA	
-		0xFFFF	
-		RW	

15 0	ADATA	Timer/Counter 20 A Data bits. The range is 0x0002 to 0xFFFF. NOTE: Do not write "0000H" in the TIMER20_ADR register when PPG mode.

14.2.3 TIMER40_BDR: Timer/counter 40 B data register

TIMER40_BDR is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_BDR=0x4000_2808

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		BDATA	
-		0xFFFF	
-		RW	

15 0	BDATA	Timer/Counter 40 B Data bits. The range is 0x0000 to 0xFFFF.

14.2.4 TIMER40_CAPDR: Timer/counter 40 capture data register

TIMER40_CAPDR is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_CAPDR=0x4000_280C

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		CAPD	
-		0x0000	
-		RO	

15 0	CAPD	Timer/Counter 40 Capture Data bits.

14.2.5 TIMER40_PREDR: Timer/counter 40 prescaler data register

TIMER40_PREDR is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_PREDR=0x4000_2810

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		PREDR	
-		0xFFF	
-		RW	

11 0	PREDR	Timer/Counter 40 Prescaler Data bits. f_{CLK}/(TIMER40_PREDR +1)

14.2.6 TIMER40_PREDR2: Timer/counter 40 prescaler data register 2

TIMER40_PREDR2 is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_PREDR2=0x4000_2814

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			PREDR2
-			0xFFF
-			RW

11 0	PREDR2	Timer/Counter 40 Capture Prescaler Data bits. $f_{Capture}$/(TIMER40_PREDR2 +1)

NOTES:

1. When using ECn (external clock) as a timer clock source in capture mode, fclk (CCR6 or PCLK) must be activated for the capture mode. This is because the fclk is used as a capture clock which is controlling the capture operation.
2. If the internal clock is used as a timer clock source, the fclk is active and the capture mode runs normally.

14.2.7 TIMER40_CNT: Timer/counter 40 counter register

TIMER40_CNT is a 16-bit register, and able to do 32/16/8-bit access.

TIMER40_CNT=0x4000_2818

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			CNT
-			0x0000
-			RO

15 0	CNT	Timer/Counter 40 Counter bits.

14.3 Functional description

14.3.1 Timer counter 40

The timer/counter 40 can be clocked by an internal or an external clock source (EC40). The clock source is selected by a clock selection logic which is controlled by the clock selection bits (T40CLK).

- TIMER 40 clock source: fCLK/(TIMER40_PREDR +1), EC40

In capture mode, by T40CAP, data is captured into input capture data register (TIMER40_CAPDR). Timer 40 outputs the comparison result between counter and data register through T40O port in timer/counter mode. In addition, Timer 40 outputs PWM wave form through T40O port in the PPG mode.

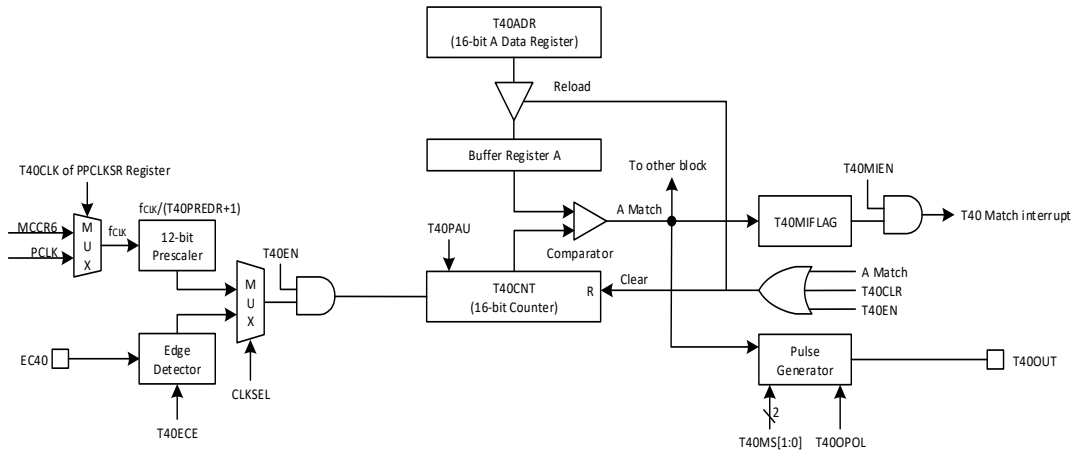
Table 62. Timer 40 Operating Modes

T40EN	Alternative Mode	T40MS[1:0]	TIMER40_PR	Timer 40 MODE
1	AF8	00	0xXXX	32-bit Timer/Counter Mode
1	AF7	01	0xXXX	32-bit Capture Mode
1	AF8	10	0xXXX	32-bit PPG Mode(one-shot mode)
1	AF8	11	0xXXX	32-bit PPG Mode(repeat mode)

14.3.2 16-bit timer/counter mode

16-bit timer/counter mode is selected by control register as shown in Figure 103. The 16-bit timer has a counter and data register. The counter register is increased by internal or external clock input. Timer 40 can use the clock input with 12-bit prescaler division rates (TIMER40_PREDR) and external clock (EC40).

When each value of TIMER40_CNT and TIMER40_ADR are identical in timer 40, a match signal is generated and the interrupt of Timer 40 occurs. The TIMER40_CNT values are automatically cleared by the match signal. It can be also cleared by software (T40CLR).



NOTES:

1. The T40EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the T40EN bit is set, the counting of TIMERn_CNT may be delayed 2 clocks or more.

Figure 103. TIMER 40 Block Diagram in 16-bit Timer/Counter Mode

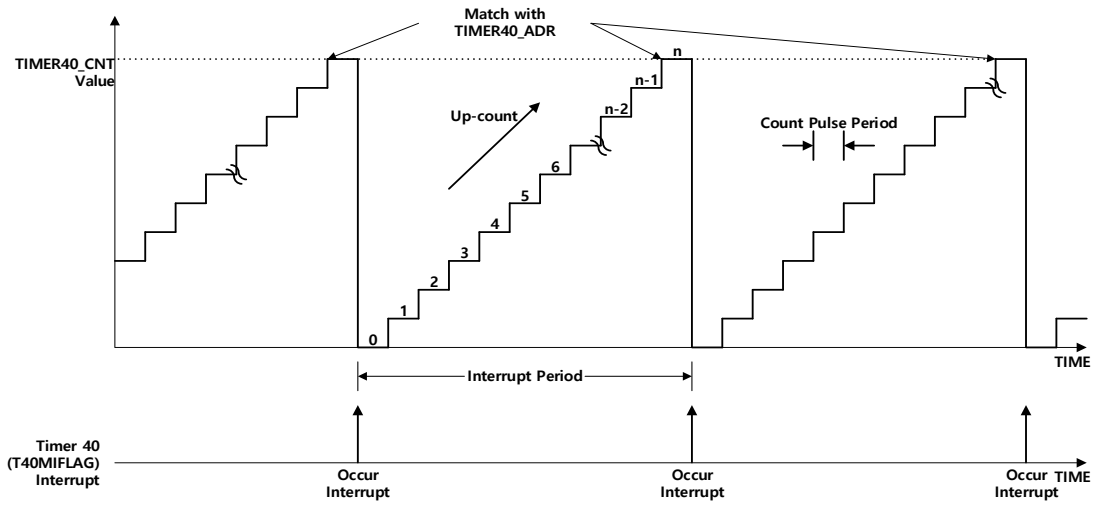


Figure 104. 16-bit Timer/Counter Mode Timing Example of TIMER 40

Figure 105 shows the timer/counter mode operation of timer 40. Refer to Figure 13 for internal timer clock source and Table 25 for Timer 40 output pin.

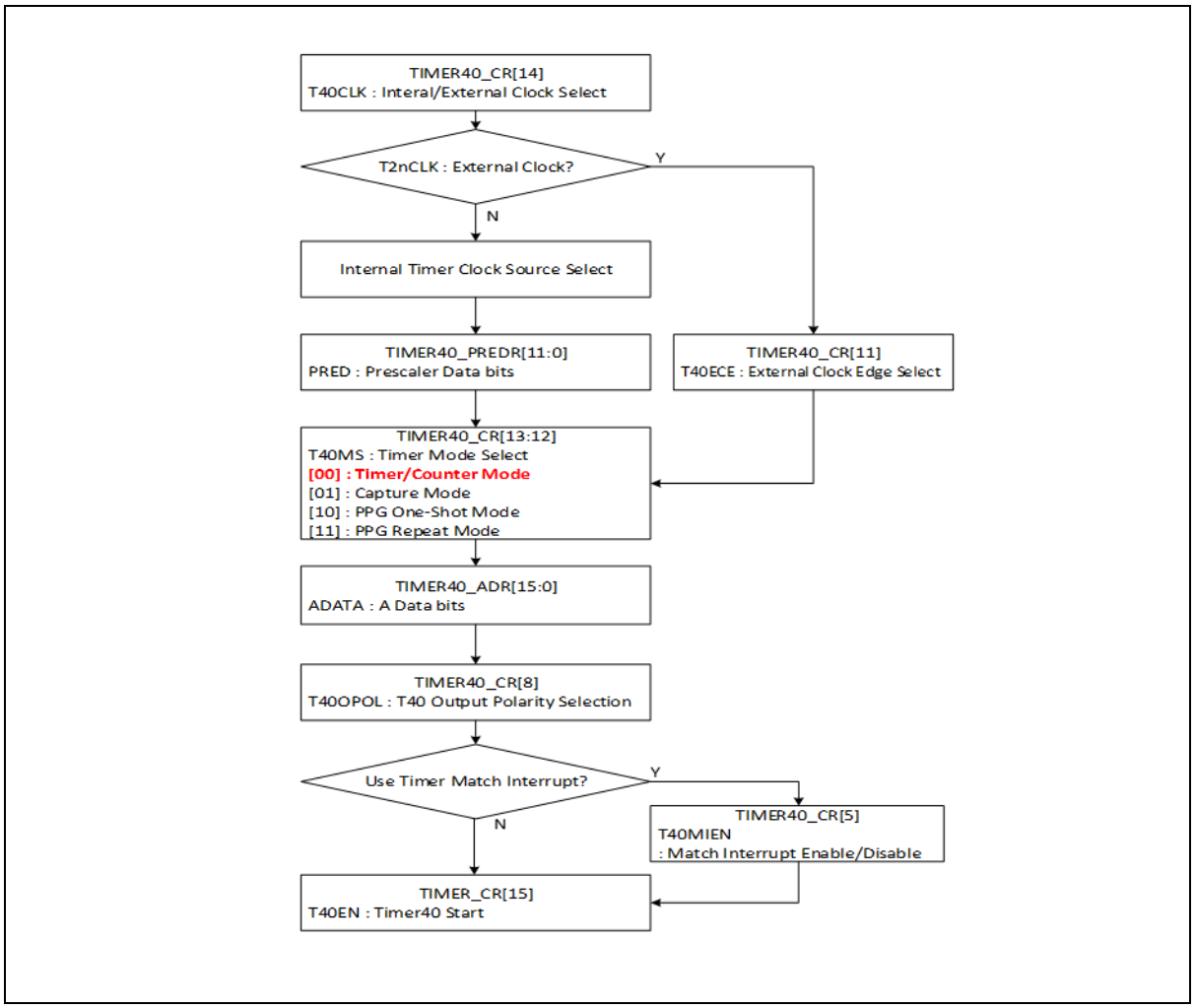


Figure 105. 16-bit Timer/Counter Mode Operation Sequence of TIMER 40

14.3.3 16-bit capture mode

TIMER40 capture mode is evoked by configuring T40MS[1:0] as '01'. The clock source can use the internal clock. Basically, it has the same function as the 32-bit timer/counter mode has, and the interrupt occurs when TIMER40_CNT is equal to TIMER40_ADR. TIMER40_CNT values are cleared by software (T40CLR).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of the timer.

The capture result is loaded into TIMER40_CAPDR. In the TIMER40 capture mode, TIMER40 output (T40O) waveform is not available.

Figure 106 shows 16-bit capture mode of the timer 40.

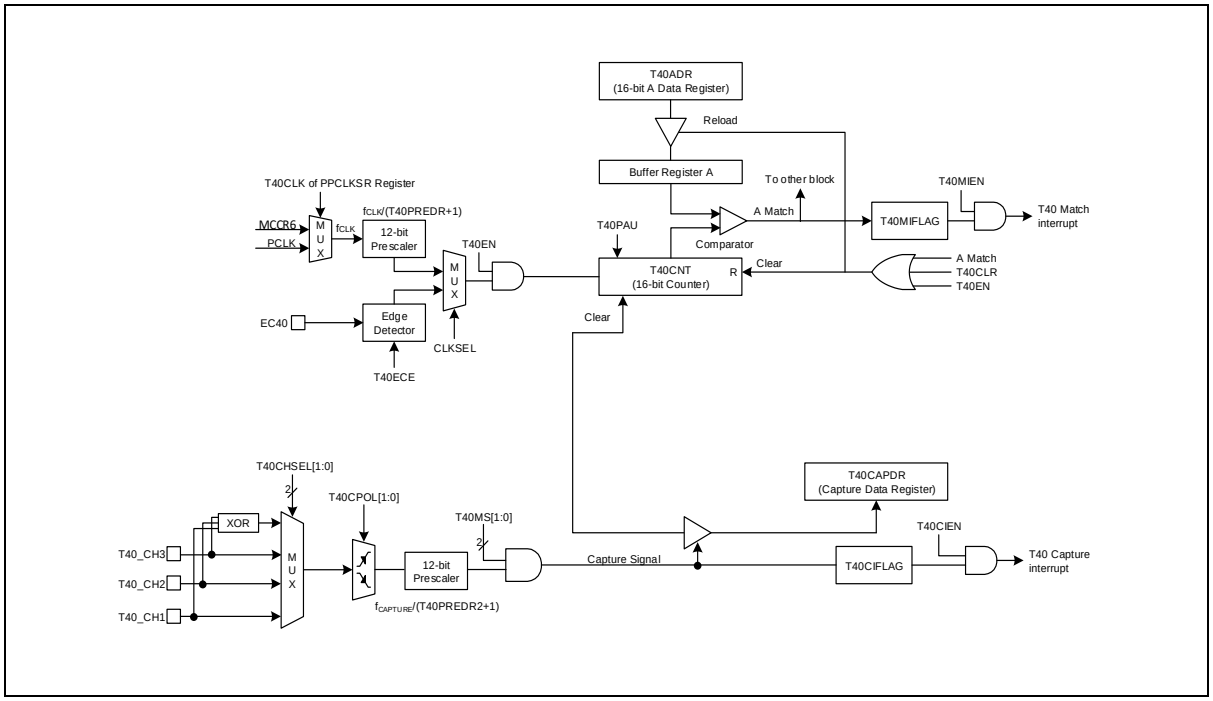


Figure 106. 16-bit Capture Mode of Timer 40

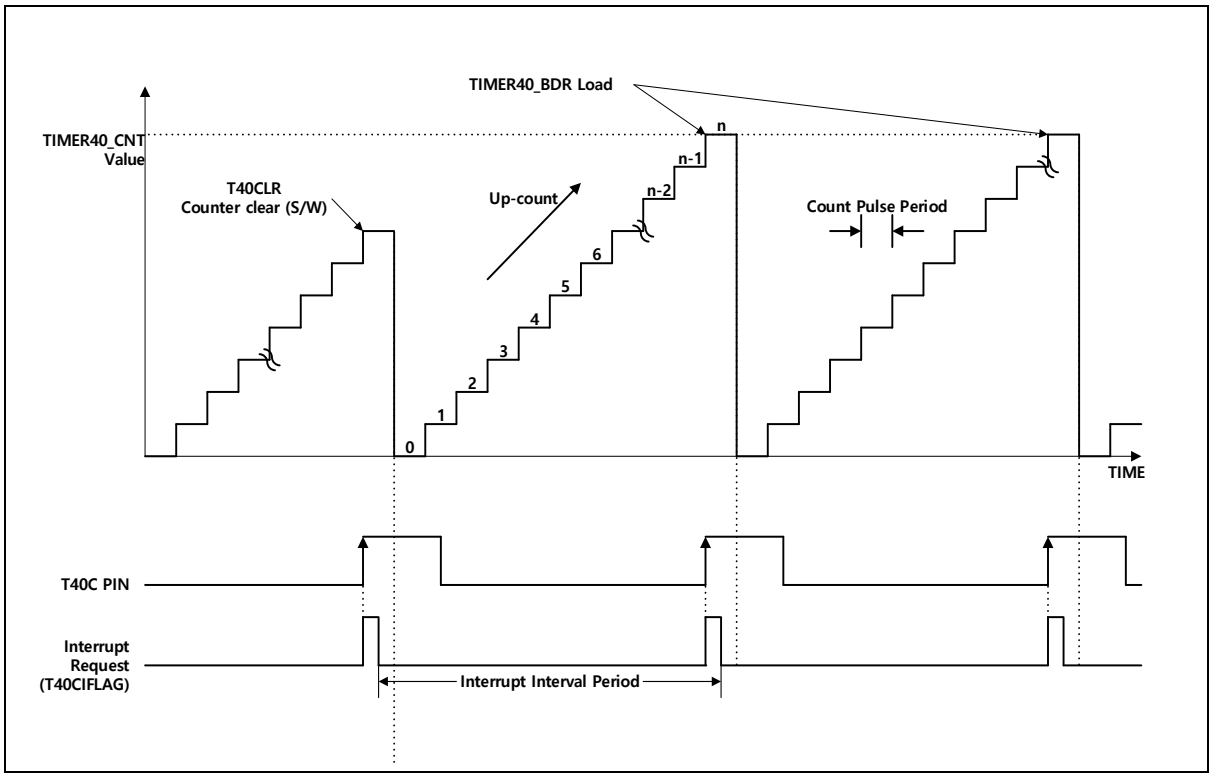


Figure 107. Capture Mode Timing Example of TIMER 40

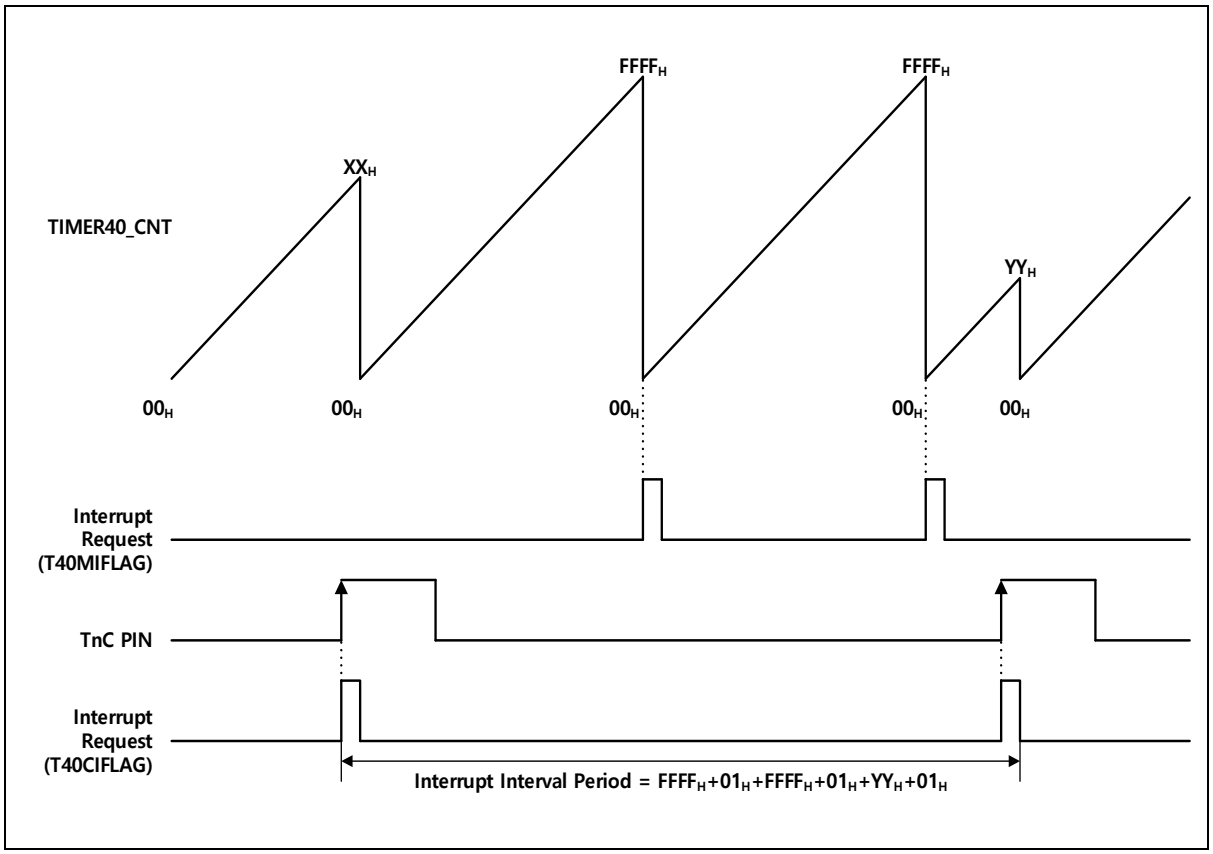


Figure 108. Express Timer Overflow in Capture Mode of TIMER 40

Figure 109 shows the capture mode operation of Timer40. Refer to Figure 13 for internal timer clock source and Table 25 for Timer 40 capture pin.

When use an external clock in TIMER40 capture mode, fclk (internal clock) must be enabled. Fclk is a Timer40 capture source.

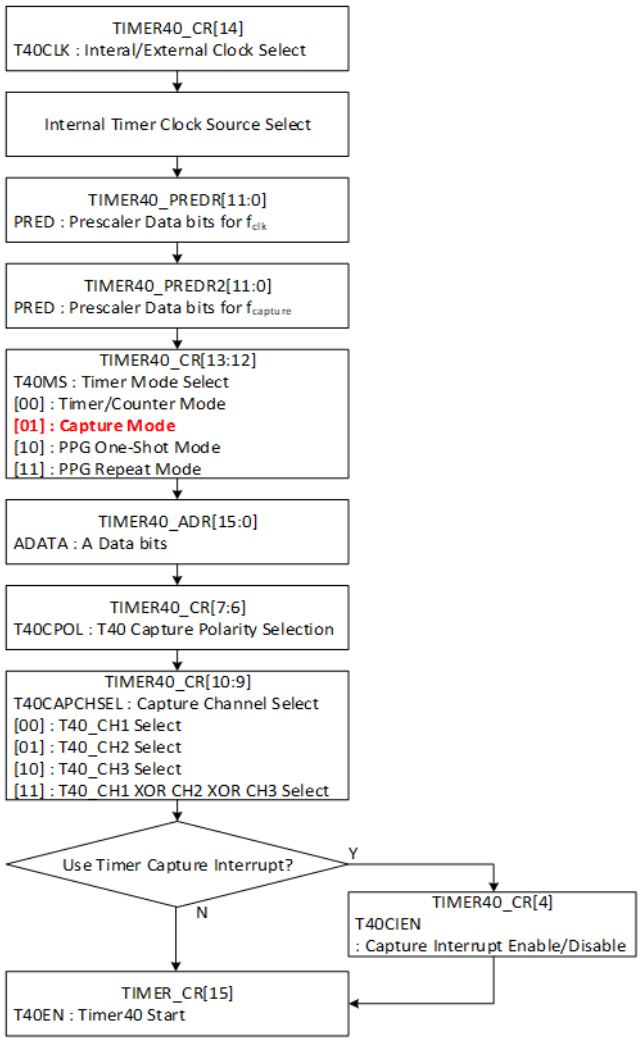
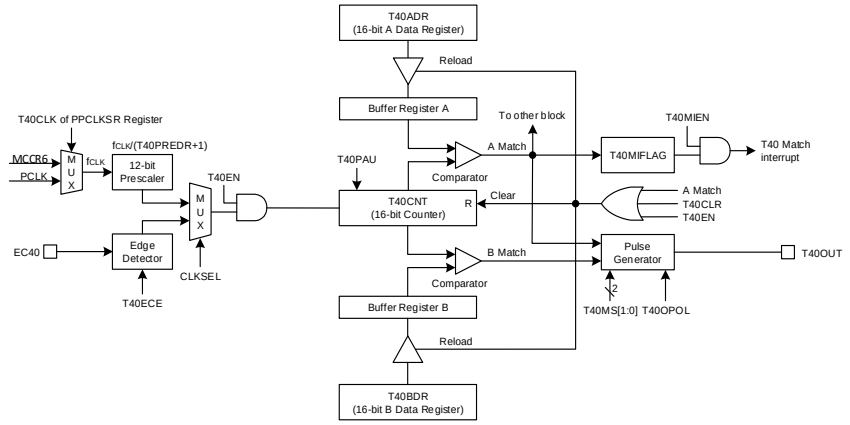


Figure 109. Capture Mode Operation Sequence of TIMER 40

14.3.4 16-bit PPG mode

Timer 40 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T40OUT pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding PnAFSRx to 'AF8'. Period of the PWM output is determined by the TIMER40_ADR, and duty of the PWM output is determined by the TIMER40_BDR.



NOTE:

1. The T40EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 110. TIMER 40 Block Diagram in PPG Mode

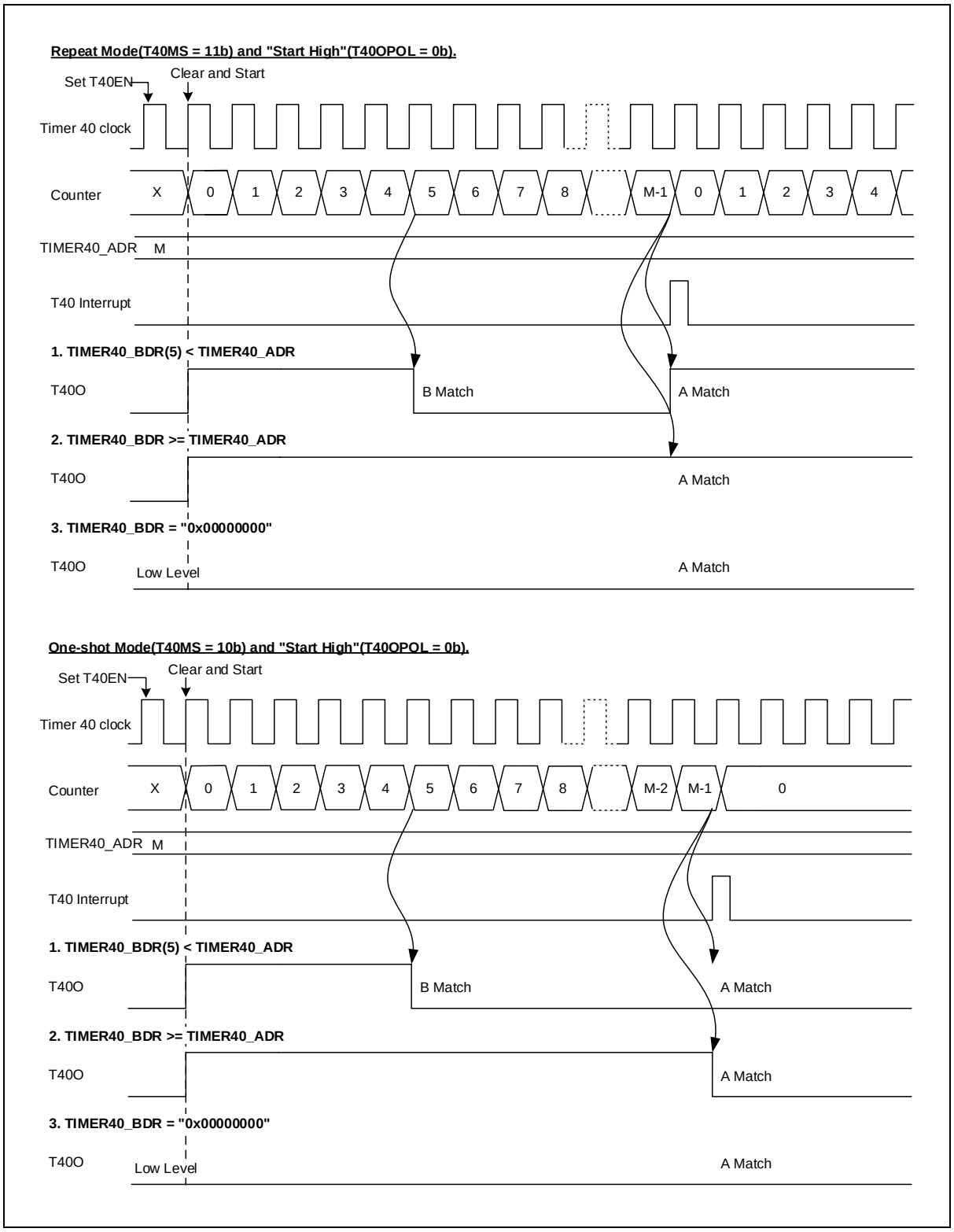


Figure 111. PPG Mode Timing Example of TIMER 40

Figure 112 shows the PPG mode operation of timer/counter 40. Refer to Figure 13 for internal timer clock source and Table 25 for Timer 40 output pin.

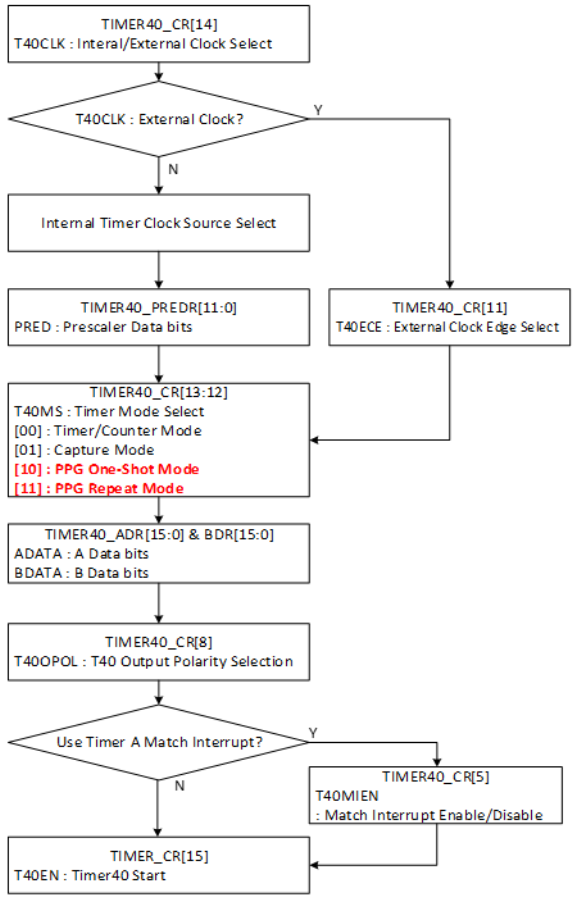


Figure 112. PPG Mode Operation Sequence of TIMER 40

15. Universal synchronous/asynchronous receiver/transmitter

Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are listed below:

- Full Duplex Operation. (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation.
- Baud Rate Generator.
- Supports Serial Frames with 5,6,7,8, or 9 Data bits and 1 or 2 Stop bits.
- Odd or Even Parity Generation and Parity Check are Supported by Hardware.
- Data OverRun Detection.
- Framing Error Detection.
- Three Separate Interrupts on TX Completion, TX Data Register Empty and RX Completion.
- Double Speed Asynchronous communication mode.

Additional features are:

- 0% Error Baud Rate by floating point count register.
- Supports receive time out interrupt.
- Supports direct memory access and interrupt.

Table 63 introduces pins assigned for the USART.

Table 63. Pin Assignment of USART: External Pins

Pin name	Type	Description
TXD	O	UART Channel n transmit output
RXD	I	UART Channel n receive input
SS	I/O	SPIn Slave select input / output
SCK	I/O	SPIn Serial clock input / output
MOSI	I/O	SPIn Serial data (Master output, Slave input)
MISO	I/O	SPIn Serial data (Master input, Slave output)

NOTE:

1. n = 10, 11, 12 and 13

15.1 USART block diagram

In this section, USART and SPIN are introduced in block diagrams.

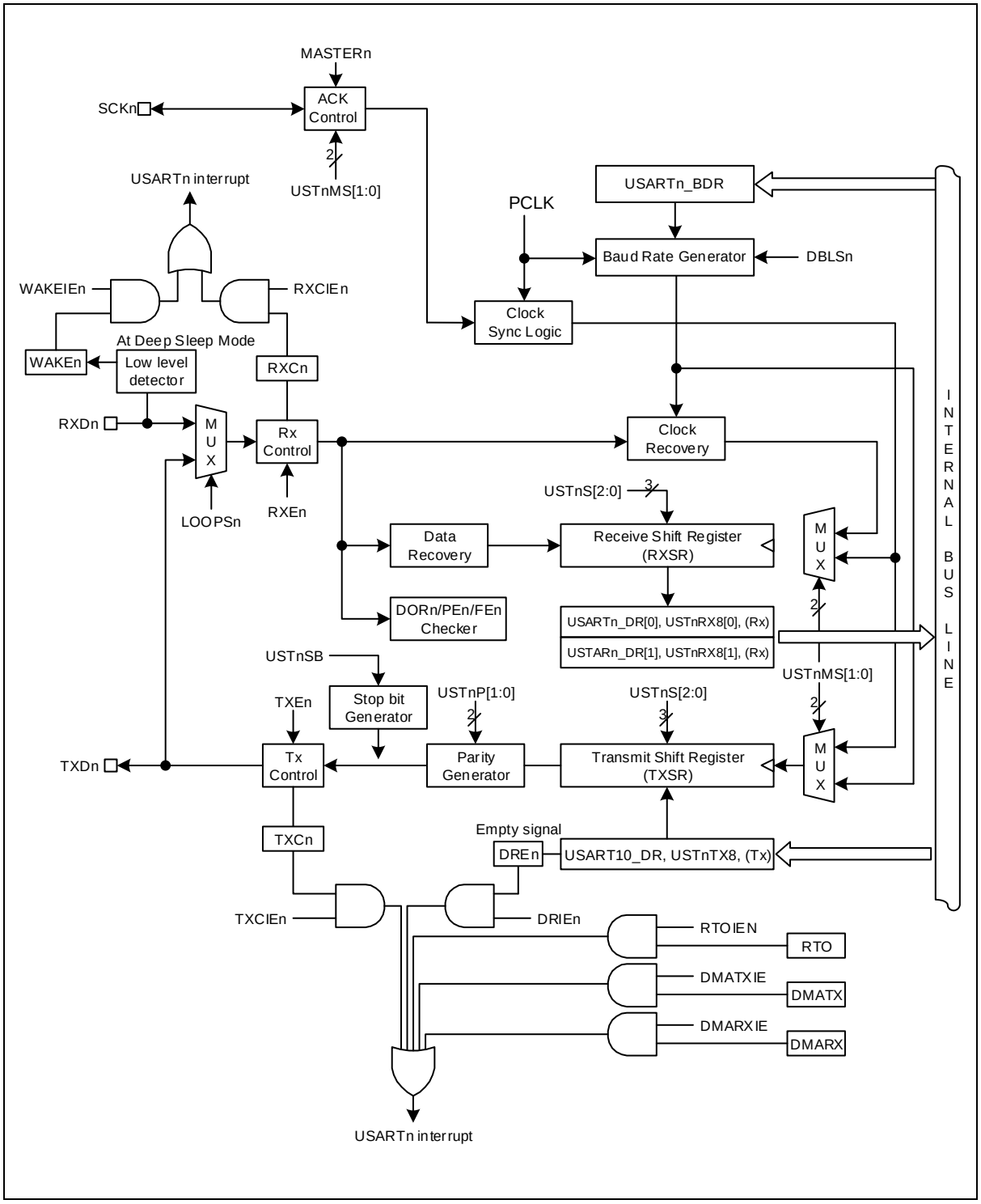


Figure 113. UART Block Diagram (n = 10, 11, 12, and 13)

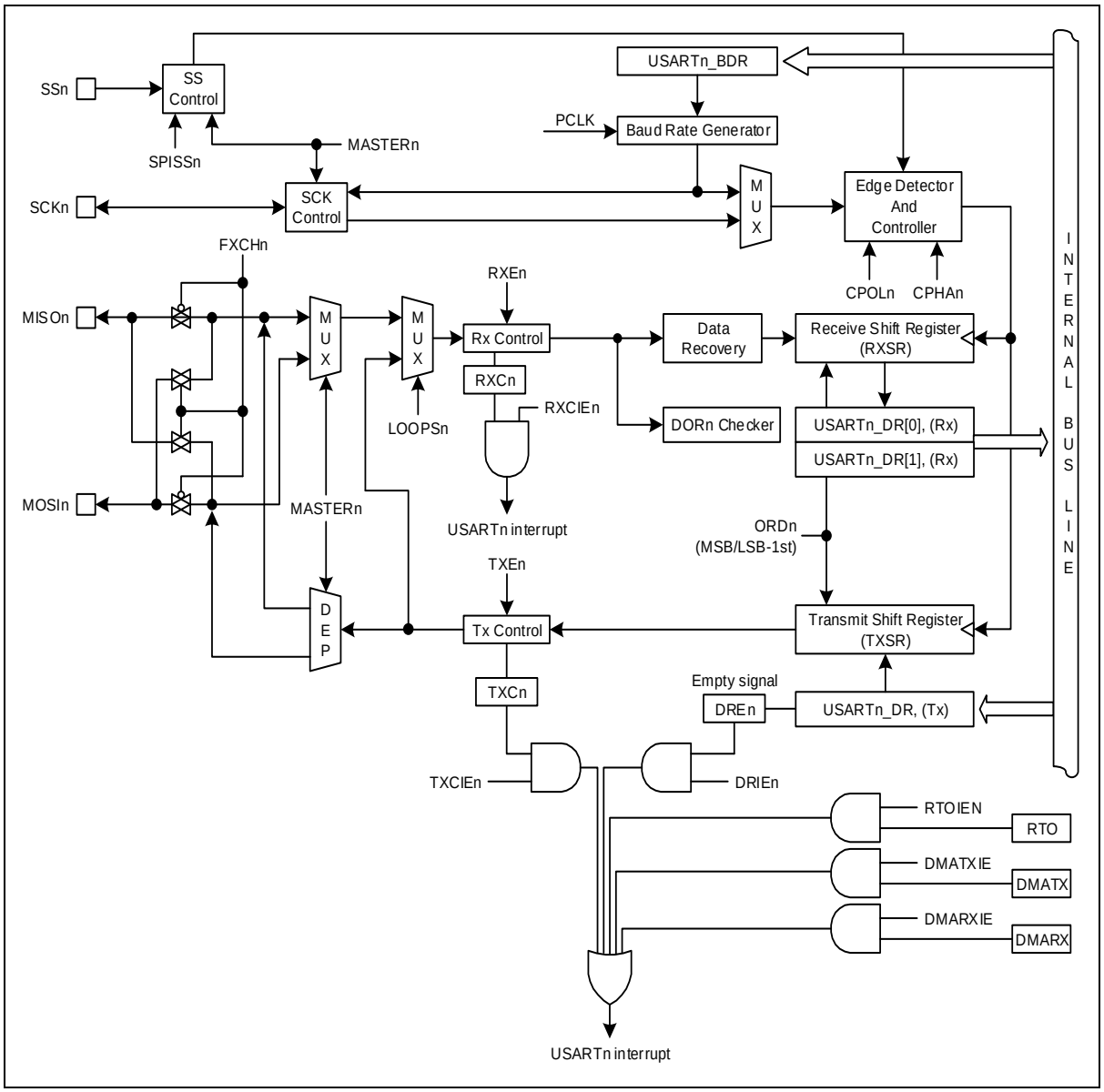


Figure 114. SPIN Block Diagram (n = 10, 11, 12, and 13)

15.2 Registers

Base address of USART is introduced in the followings:

Table 64. Base Address of USART

Name	Base address
USART 10	0x4000_3800
USART 11	0x4000_3900
USART 12	0x4000_3A00
USART 13	0x4000_3B00

Table 65. USART Register Map

Name	Offset	Type	Description	Reset value	Reference
USARTn_CR1	0x00	RW	USARTn Control Register 1	0x0000_0000	15.2.1
USARTn_CR2	0x04	RW	USARTn Control Register 2	0x0000_0000	15.2.2
USARTn_ST	0x0C	RW	USARTn Status Register	0x0000_0080	15.2.3
USARTn_BDR	0x10	RW	USARTn Baud Rate Generation Register	0x0000_0FFF	15.2.4
USARTn_DR	0x14	RW	USARTn Data Register	0x0000_0000	15.2.5
USARTn_FPCR	0x18	RW	Floating Point Count Register	0x0000_0000	15.2.6
USARTn_RTO	0x1C	RW	RTO Register	0x0000_0000	15.2.7

NOTE:

1. n = 10, 11, 12 and 13

15.2.1 USARTn_CR1: USARTn control register 1

USART module should be configured properly before running. USARTn_CR1 is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_CR1=0x4000_3800, USART11_CR1=0x4000_3900
USART12_CR1=0x4000_3A00, USART13_CR1=0x4000_3B00

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		USTnMS		USTnP		USTnS			ORDn	CPOLn	CPHAn	DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn
-		00		00		000			0	0	0	0	0	0	0	0	0
-		RW		RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW

15 14	USTnMS	USARTn Operation Mode Selection bits.	
		00	Asynchronous Mode. (UART)
		01	Synchronous Mode.
		10	Reserved.
		11	SPI Mode
13 12	USTnP	Selects Parity Generation and Check method. (only UART mode)	
		00	No parity.
		01	Reserved.
		10	Even parity.
		11	Odd parity.
11 9	USTnS	Selects the length of data bit in a frame when Asynchronous or Synchronous mode.	
		000	5 bit.
		001	6 bit.
		010	7 bit.
		011	8 bit.
		111	9 bit.
		Others	Reserved.

8	ORDn	Selects the first data bit to be transmitted. (only SPI mode)			
		0	LSB-first.		
		1	MSB-first.		
7	CPOLn	Selects the clock polarity of ACK in synchronous or SPI mode.			
		0	TXD Change @Rising Edge, RXD Change @Falling Edge.		
		1	TXD Change @Falling Edge, RXD Change @Rising Edge.		
6	CPHAn	The CPOLn and this bit determine if data are sampled on the leading or trailing edge of SCK. (only SPI mode)			
		CPOLn	CPHAn	Leading edge	Trailing edge
		0	0	Sample (Rising)	Setup (Falling)
		0	1	Setup (Rising)	Sample (Falling)
		1	0	Sample (Falling)	Setup (Rising)
		1	1	Setup (Falling)	Sample (Rising)
5	DRIEn	Transmit Data Register Empty Interrupt Enable bit.			
		0	Disable the transmit data empty interrupt.		
		1	Enable the transmit data empty interrupt.		
4	TXCIEn	Transmit Complete Interrupt Enable bit.			
		0	Disable transmit complete interrupt.		
		1	Enable transmit complete interrupt.		
3	RXCIEn	Receive Complete Interrupt Enable bit.			
		0	Disable receive complete interrupt.		
		1	Enable receive complete interrupt.		
2	WAKEIEn	Asynchronous Wake-up Interrupt Enable bit in Deep Sleep Mode. When device is in deep sleep mode, if RXDn goes to low level, an interrupt can be requested to wake-up system. (only UART mode) Must be set "1b" in Stop mode, set to "0b" if wake up from stop mode.			
		0	Disable asynchronous wake-up interrupt.		
		1	Enable asynchronous wake-up interrupt.		
1	TXEn	Enables the Transmitter unit.			
		0	Transmitter is disabled.		
		1	Transmitter is enabled.		
0	RXEn	Enables the Receiver unit.			
		0	Receiver is disabled.		
		1	Receiver is enabled.		
NOTE: The CPOLn and CPHAn bits should be changed during the TXEn and RXEn bits are "0b"					

15.2.2 USARTn_CR2: USARTn control register 2

USART module should be configured properly before running. USARTn_CR2 is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_CR2=0x4000_3804, USART11_CR2=0x4000_3904
USART12_CR2=0x4000_3A04, USART13_CR2=0x4000_3B04

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			DMATXIE	DMARXIE	RTOIE	RTOEN	FPCREN	USTnEN	DBLSn	MASTERn	LOOPSn	DISSCKn	USTnSSEN	FXCHn	USTnSB	USTnTX8	USTnRX8
-			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

14	DMATXIE	DMA TX Interrupt bit	
		0	Disable DMA TX Interrupt
		1	Enable DMA TX Interrupt
13	DMARXIE	DMA RX Interrupt bit	
		0	Disable DMA RX Interrupt
		1	Enable DMA RX Interrupt
12	RTOIE	RTO Interrupt bit	
		0	Disable RTO Interrupt
		1	Enable RTO Interrupt
11	RTOEN	Activate RTO Block by supplying. After RTO occurs, Clear.	
		0	Disable RTO
		1	Enable RTO
10	FPCREN	Activate Floating Point Counter Register	
		0	Disable Floating Point Counter Register
		1	Enable Floating Point Counter Reigster.
9	USTnEN	Activate USARTn Block by supplying.	
		0	Disable USARTn block.
		1	Enable USARTn block.
8	DBLSn	Selects receiver sampling rate. (only UART mode)	
		0	Normal asynchronous operation.
		1	Double speed asynchronous operation.
7	MASTERn	Selects master or slave in SPIn or synchronous mode and controls the direction of SCKn pin.	
		0	Slave operation. (External clock for SCKn)
		1	Master operation. (Internal clock for SCKn)
6	LOOPSn	Control the Loop Back mode of USARTn for test mode.	
		0	Normal operation.
		1	Loop Back mode.
5	DISSCKn	In synchronous mode operation, selects the waveform of SCKn output.	
		0	SCKn is free-running while USARTn is enabled in synchronous master mode.
		1	SCKn is active while any frame is on transferring.

4	USTnSSEN	This bit controls the SSn pin operation. (only SPI mode)	
		0	Disable.
		1	Enable.
3	FXCHn	SPIn port function exchange control bit. (only SPI mode)	
		0	No effect.
		1	Exchange MOSIn and MISOn function.
2	USTnSB	Selects the length of stop bit in asynchronous or synchronous mode.	
		0	1 Stop bit.
		1	2 Stop bit.
1	USTnTX8	The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USTnDR register.	
		0	MSB (9th bit) to be transmitter is '0'.
		1	MSB (9th bit) to be transmitter is '1'.
0	USTnRX8	The ninth bit of data frame in asynchronous or synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode)	
		0	MSB (9th bit) to be received is '0'.
		1	MSB (9th bit) to be received is '1'.

15.2.3 USARTn_ST: USARTn status register

USARTn_ST is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_ST =0x4000_380C, USART11_ST =0x4000_390C
USART12_ST =0x4000_3A0C, USART13_ST =0x4000_3B0C

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			DMATXF	DMARXF	DREn	TXCn	RXCn	WAKEn	RTOF	DORn	Fen	Pen
-			0	0	1	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RO	RW	RW	RO	RW	RW

9	DMATXF	DMA Transmit Operation Complete flag. (DMA to USART)	
		0	DMA Transmit Operation is working or is disabled
		1	DMA Transmit Op is done
8	DMARXF	DMA Receive Operation Complete flag. (USART to DMA)	
		0	DMA Receive Operation is working or is disabled
		1	DMA Transmit Op is done
7	DREn	Transmit Data Register Empty Interrupt Flag. The DRE flag indicates if the transmit data register (USARTn_DR) is ready to receive new data. If DRE is '1', the data register is empty and ready to be written.	
		0	Transmit buffer is not empty.
		1	Transmit buffer is empty. This bit is cleared to '0' when write '1'.
6	TXCn	Transmit Complete Interrupt Flag. This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer.	
		0	No request occurred.
		1	Transmit buffer is empty and the data in transmit shift register are shifted out completely. This bit is cleared to '0' when write '1'.

5	RXCn	Receive Complete Interrupt Flag. This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read.	
		0	There is no data unread in the receive buffer.
		1	There are more than 1 data in the receive buffer.
4	WAKEn	Asynchronous Wake-up Interrupt Flag. This flag is set when the RXD pin is detected low while the CPU is in deep sleep mode. (only UART mode)	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3	RTOF	Receive Time Out Interrupt flag. This bit is cleared to '0' when write '1'.	
		0	Receive time out is not generated
		1	Receive time out is generated
2	DORn	This bit is set if data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.	
		0	No Data OverRun.
		1	Data OverRun detected.
1	Fen	This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read. (only UART mode)	
		0	No Frame Error.
		1	Frame Error detected.
0	Pen	This bit is set if the next character in the receive buffer has a Parity Error while parity is checked. This bit is valid until the receive buffer is read. (only UART mode)	
		0	No Parity Error.
		1	Parity Error detected.

15.2.4 USARTn_BDR: USARTn baud rate generation register

USARTn_BDR is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_BDR =0x4000_3810, USART11_BDR =0x4000_3910
USART12_BDR =0x4000_3A10, USART13_BDR =0x4000_3B10

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			BDATA
-			0xFFF
-			RW

11 0	BDATA	The value in this register is used to generate internal baud rate in UART mode or to generate SCK clock in SPI mode. To prevent malfunction, do not write '0' in UART mode and do not write '0' or '1' in synchronous or SPI mode.

15.2.5 USARTn_DR: USARTn data register

USARTn_DR is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_DR =0x4000_3814, USART11_DR =0x4000_3914
USART12_DR =0x4000_3A14, USART13_DR =0x4000_3B14

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			DATA
-			0x00
-			RW

7 0	DATA	The USART Transmit buffer and Receive buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the USARTn_DR register. Reading the USTnDR register returns the contents of the Receive Buffer. Write to this register only when the DRE flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

NOTE: This byte won't be written when the block is disabled or the both of TXEn and RXEn bits are "0b".

15.2.6 USARTn_FPCR: USARTn floating point count register

USARTn_FPCR is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_FPCR =0x4000_3818, USART11_ FPCR =0x4000_3918
USART12_ FPCR =0x4000_3A18, USART13_ FPCR =0x4000_3B18

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			FPCR
-			0xFFF
-			RW

7 0	FPCR	USART Floating Point Counter 8-bit floating point counter

15.2.7 USARTn_RTO: USARTn RTO register

USARTn_RTO is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11, 12, and 13).

USART10_RTO =0x4000_381C, USART11_ RTO =0x4000_391C
USART12_ RTO =0x4000_3A1C, USART13_ RTO =0x4000_3B1C

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	RTO		
-	0x000000		
-	RW		

23 0	RTO	USART receive time out register

15.3 Functional description

USART comprises clock generator, transmitter and receiver.

The clock generation logic consists of synchronization logic for external clock input used by synchronizing or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation.

The transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames.

The receiver is the most complex part of the UART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFO (USARTn_DR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors (n = 10, 11, 12 and 13).

15.3.1 USART clock generation

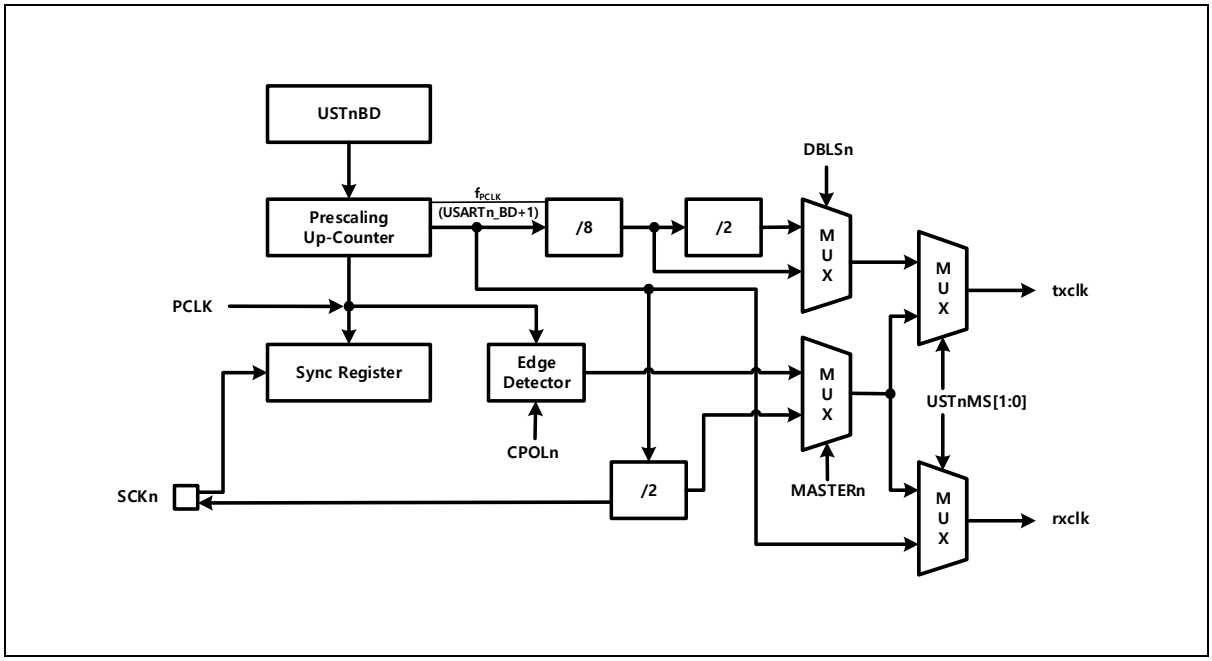


Figure 115. Clock Generation Block Diagram (USARTn, n = 10, 11, 12, and 13)

The clock generation logic generates the base clock for the transmitter and receiver. The USART supports four modes of clock operation and those are normal asynchronous, double speed asynchronous, master synchronous and slave synchronous modes.

The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode.

USTnMS[1:0] bits in USARTn_CR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLS bit in the USARTn_CR2 register.

The MASTER bit in USARTn_CR2 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USART operates in synchronous or SPI mode.

Table 67 shows the equations for calculating baud rate (in bps).

Table 66. Equations for Calculating USART Baud Rate Register Settings

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (DBLSn=0)	Baud Rate= PCLK/(16(USARTn_BDR+1))
Asynchronous Double Speed Mode (DBLSn=1)	Baud Rate= PCLK/(8(USARTn_BDR+1))
Synchronous or SPI Master Mode	Baud Rate= PCLK/(2(USARTn_BDR+1))

NOTE:

1. n = 10, 11, 12 and 13

15.3.2 External clock (SCKn)

External clocking is used in the synchronous or SPI slave mode of operation.

External clock input from the SCKn pin is sampled by a synchronization logic to remove meta-stability. Output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and receiver. This process introduces two CPU clock period delay. The maximum frequency of the external SCKn pin is limited by 1MHz.

15.3.3 Synchronous mode operation

External clocking is used in the synchronous or SPI slave mode of operation.

When synchronous or SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter are issued on the different edge of SCKn clock respectively. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

CPOLn bit in USARTn_CR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in the Figure 116, when CPOLn is zero, the data will be changed at rising edge of SCKn and sampled at falling edge of SCKn.

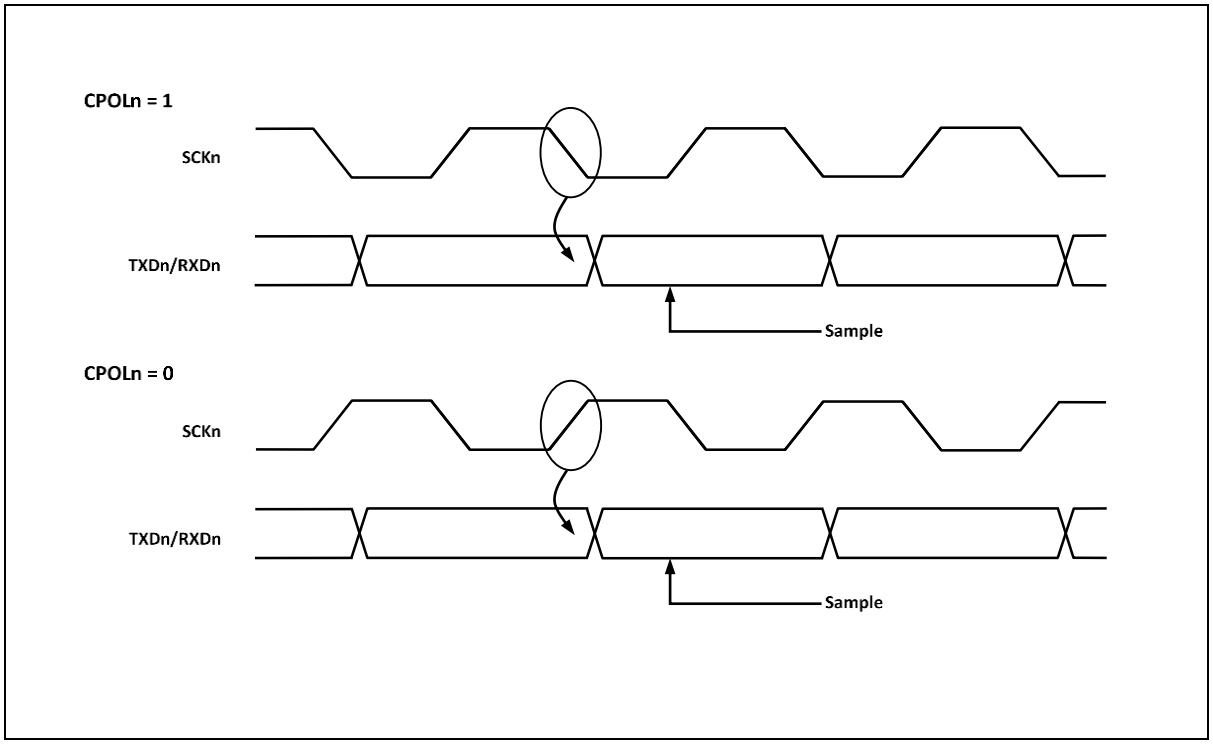


Figure 116. Synchronous Mode SCKn Timing (USARTn, n = 10, 11, 12, and 13)

15.3.4 UART data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection. USART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit.
- 1 or 2 stop bits.

A frame starts with a start bit followed by the least significant data bit (LSB). Next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit.

A high-to-low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin.

Figure 117 shows possible combinations of the frame formats. Bits inside brackets are optional.

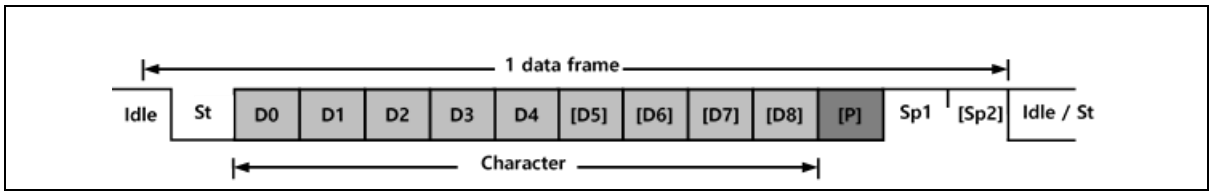


Figure 117. Frame Format (USART)

Single data frame consists of the following bits:

- Idle: No communication on communication line (TXDn/RXDn)
- St: Start bit (Low)
- Dm: Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

Frame format of UART is set by configuring USTnS[2:0], USTnP[1:0] bits in USARTn_CR1 register and USTnSB bit in USARTn_CR2 register. Transmitter and receiver use the same figures (n = 10, 11, 12 and 13).

15.3.5 UART parity bit

Parity bit is calculated by doing an exclusive-OR of all data bits. If odd parity is used, the result of the exclusive-OR is inverted. Parity bit is located between the MSB and first stop bit of a serial frame.

- Peven = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 0
- Podd = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 1
- Peven: Parity bit using even parity
- Podd: Parity bit using odd parity
- Dm: Data bit n of the character

15.3.6 UART transmitter

UART transmitter is enabled by configuring TXEn bit in USARTn_CR1 register. When the transmitter is enabled, TXDn pin should be set to TXDn function for the serial output pin of UART by the PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. Baud rate, operation mode and frame format must be set up once before starting any transmission.

In synchronous operation mode, SCKn pin is used as a transmission clock, so it should be selected to function SCKn by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD (n = 10, 11, 12 and 13).

UART sending TX data

A data transmission is initiated by loading the transmit buffer (USARTn_DR register I/O location) with the data to be transmitted. The data be written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode, the ninth bit must be written to the USTnTX8 bit in USARTn_CR2 register before it is loaded to the transmit buffer USARTn_DR register (n = 10, 11, 12 and 13).

UART transmitter flag and interrupt

The USART transmitter has two flags which indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion flag (TXCn). Both flags can be interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing '0' to this bit position. Writing '1' to this bit position is prevented.

When the data register empty interrupt enable (DRIEn) bit in USARTn_CR1 register is set and the global interrupt is enabled, USTnST status register empty interrupt is generated while DREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt serve routine is executed, or it can be cleared by writing '0' to TXCn bit in USARTn_ST register.

When the transmit complete interrupt enable (TXCIEn) bit in USARTn_CR1 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXCn flag is set (n = 10, 11, 12 and 13).

UART parity generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USTnP1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

UART disabling transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (n = 10, 11, 12 and 13).

15.3.7 UART receiver

USART receiver is enabled by setting the RXEn bit in the USARTn_CR1 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin of UART by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCKn pin is used as transfer clock input, so it should be selected to do SCKn function by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. In SPI operation mode the SSn input pin in slave mode can be configured as SSn output pin in master mode. This can be done by setting USTnSSEN bit in USARTn_CR2 register (n = 10, 11, 12 and 13).

UART receiving RX data

When UART is in synchronous or asynchronous operation mode, the receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCKn (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's the second stop bit in the frame, the second stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is presented in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the USARTn_DR register.

If 9-bit characters are used (USTnS[2:0] = "111"), the ninth bit is stored in the USTnRX8 bit position in the USARTn_CR2 register. The ninth bit must be read from the USTnRX8 bit before reading the low 8 bits from the USARTn_TDR register. Likewise, the error flags Fen, DORn, Pen must be read before reading the data from USARTn_DR register. It's because the error flags are stored in the same FIFO position of the receive buffer (n = 10, 11, 12 and 13).

UART receiver flag and interrupt

The UART receiver has one flag that indicates the receiver state.

A receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there is unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When a receive complete interrupt enable (RXCIEn) bit in the USARTn_CR1 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags which are frame error (Fen), data overrun (DORn) and parity error (Pen). These error flags can be read from the USARTn_ST register. As received data is stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from USARTn_DR register, read the USARTn_ST register first which contains error flags.

The frame error (Fen) flag indicates the state of the first stop bit. The Fen flag is "0" when the stop bit was correctly detected as "1", and the Fen flag is "1" when the stop bit was incorrect, i.e. detected as "0". This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is presented in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To avoid data loss or clear this flag, read the receive buffer.

The parity error (Pen) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USTnP1=0), the Pen bit is always read "0" (n = 10, 11, 12 and 13).

UART parity checker

If parity bit is enabled (USTnP1=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame (n = 10, 11, 12 and 13).

UART disabling receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately.

When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (n = 10, 11, 12 and 13).

Asynchronous data reception

To receive asynchronous data frame, the USART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin.

The data recovery logic does sampling and low pass filtering the incoming bits, and removing the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode (n = 10, 11, 12 and 13).

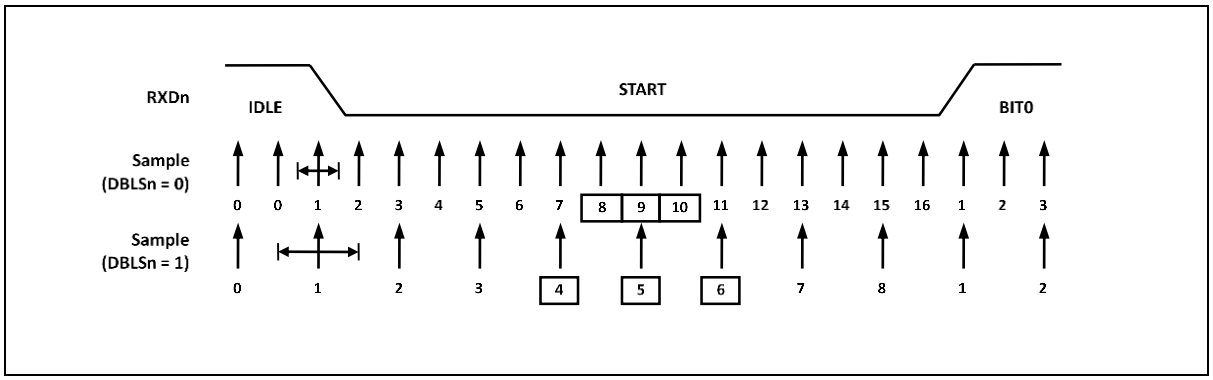


Figure 118. Asynchronous Start Bit Sampling (n = 10, 11, 12 and 13)

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost same to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic '0' and if more than 2 samples have high levels, the received bit is considered to a logic '1'.

The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

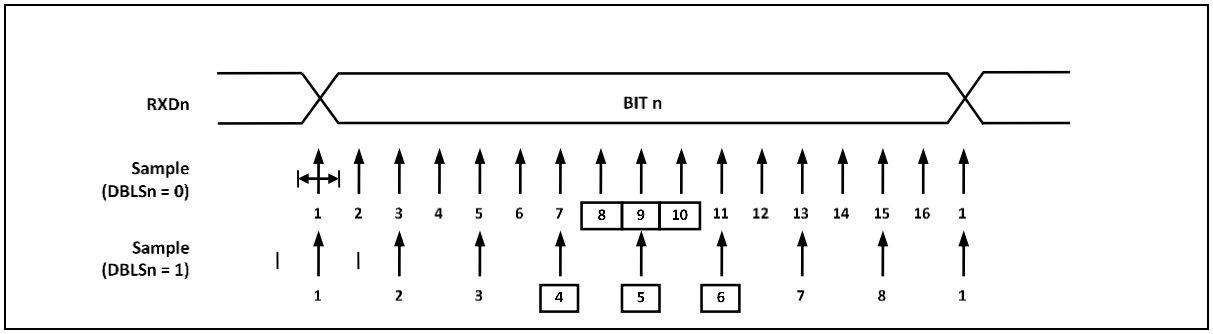


Figure 119. Asynchronous Data and Parity Bit Sampling (n = 10, 11, 12 and 13)

The process for detecting stop bit is same as clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (Fen) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXDn line to check a valid high to low transition is detected (start bit detection). (n = 10, 11, 12 and 13).

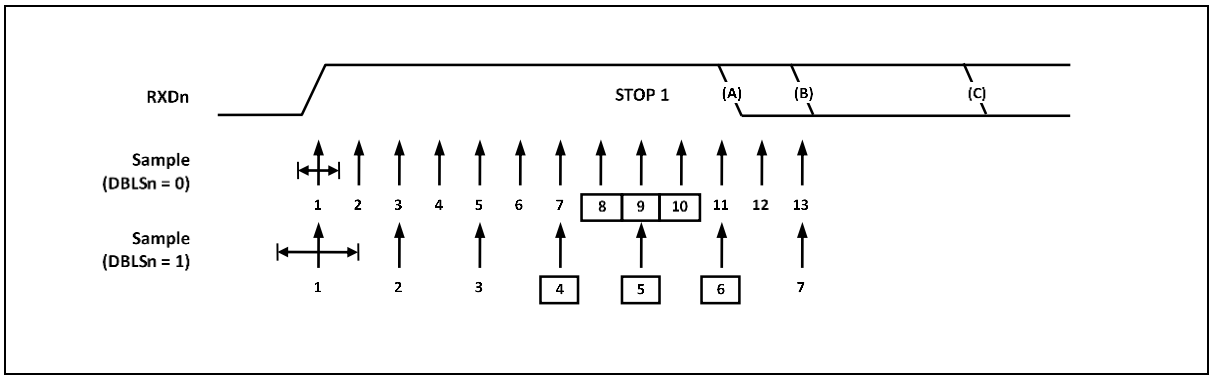


Figure 120. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11, 12 and 13)

15.3.8 SPI mode

USART can be set to operate in industrial standard SPI compliant mode. The SPI mode features the followings:

- Full Duplex, Three-wire synchronous data transfer.
- Mater and Slave Operation.
- Supports all four SPI modes of operation (mode 0, and 1).
- Selectable LSB first or MSB first data transfer.
- Double buffered transmit and receive.
- Programmable transmit bit rate.

When the SPI mode is enabled by configuring USTnMS[1:0] as "11", the slave select (SSn) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USTnSSEN bit is set to '0'.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn, and TXDn is renamed as MOSIn for compatibility to other SPI devices (n = 10, 11, 12, and 13).

15.3.9 SPI clock formats and timing

To accommodate a wide variety if synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively insert an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USARTn_CR1 register have different meanings according to the USTnMS[1:0] bits which decides the operating mode of USART.

Table 67 shows four combinations of CPOLn and CPHAn for SPI modes 10, 11, 12 and 13 (n = 10, 11, 12, and 13).

Table 67. CPOL Functionality

SPIn Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

NOTE:

1. n = 10, 11, 12 and 13

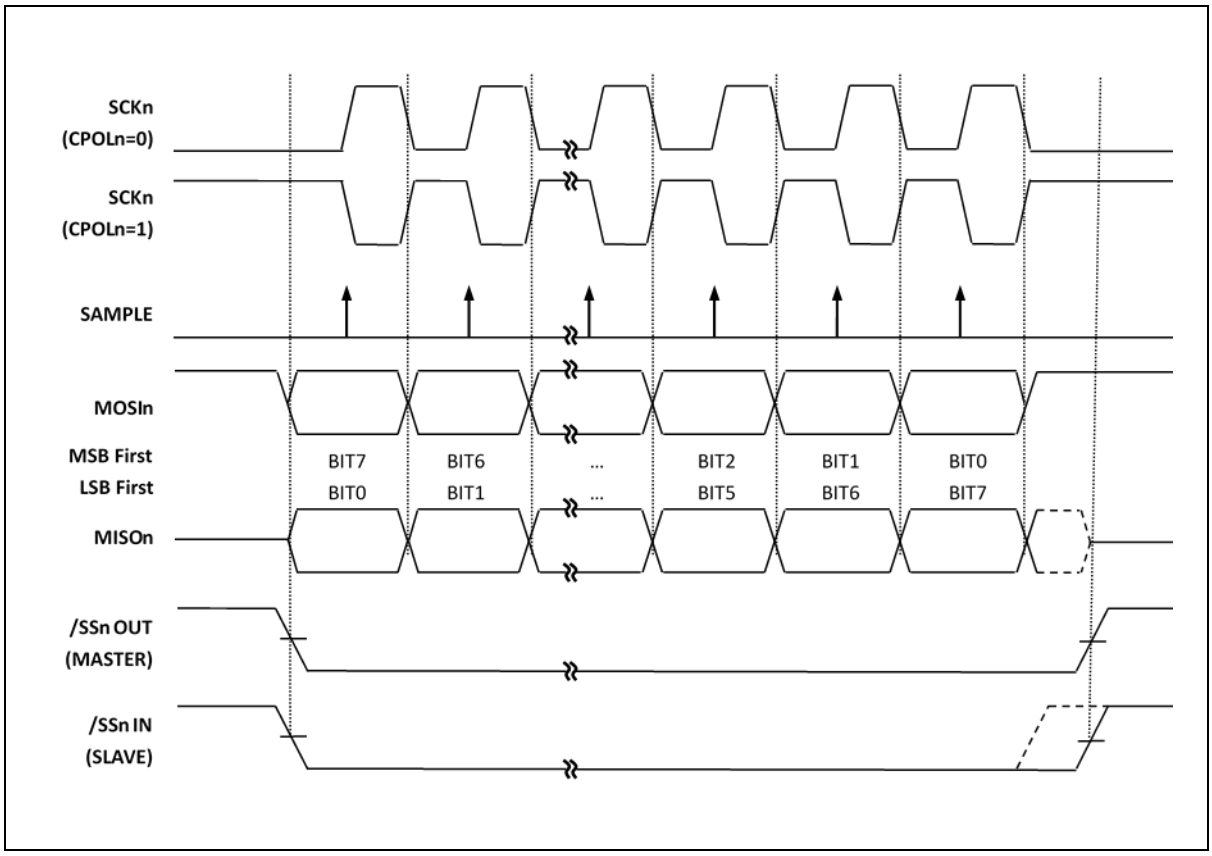


Figure 121. USART SPIn Clock Formats when CPHAn = 0 (n = 10, 11, 12 and 13)

When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the second SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. Unlike the case of CPHAn=1, when CPHAn=0, the slave's SSn input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SSn input (n = 10, 11, 12 and 13).

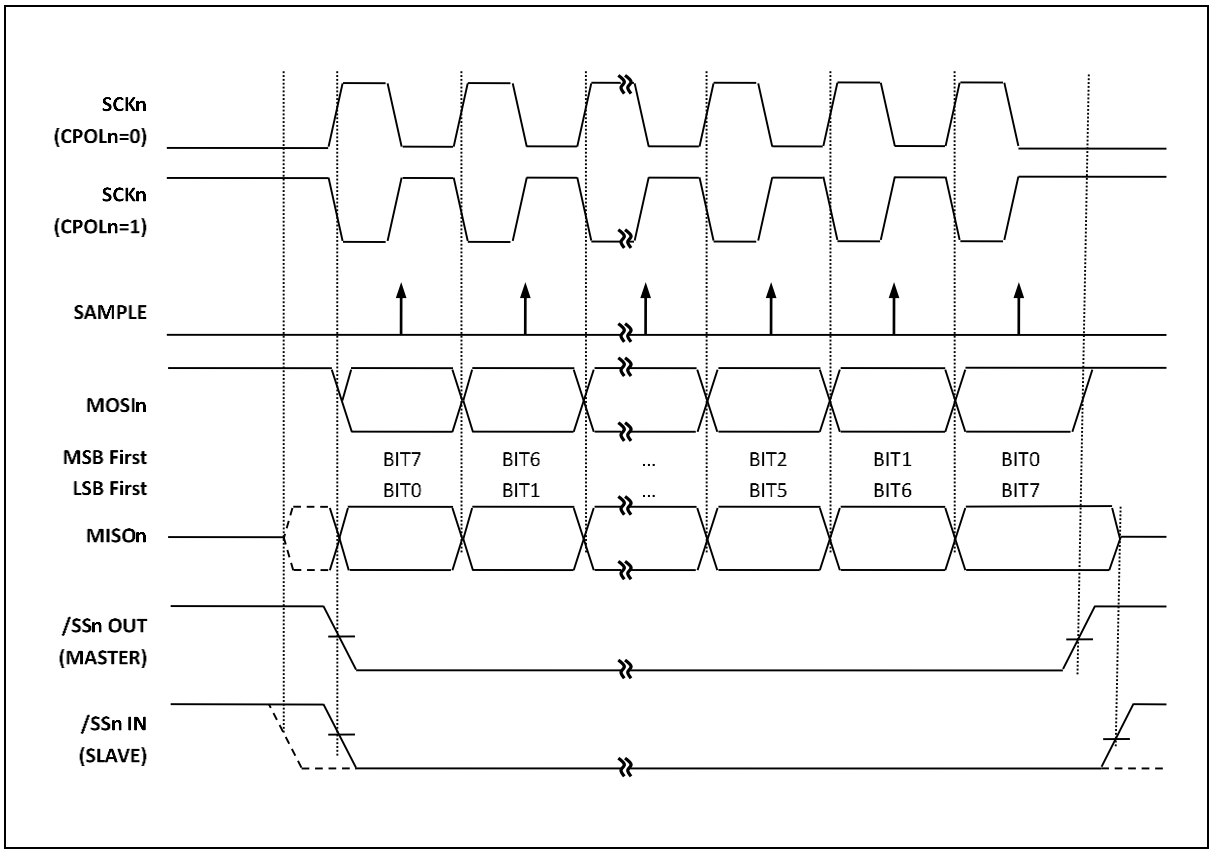


Figure 122. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11, 12 and 13)

When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge. The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave.

The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the third SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively. When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

Because the SPIn logic reuses the USART resources, SPIn mode of operation is similar to that of synchronous or asynchronous operation. SPIn transfer is initiated by checking for the USART Data Register Empty flag (DREn=1) and then writing a byte of data to the USTnDR Register. In master mode of operation, even if transmission is not enabled (TXEn=0), writing data to the USTnDR register is necessary because the clock SCKn is generated from transmitter block.

15.3.10 Baud rate settings (example)

Table 68. Baud Rate Settings Example

Baud Rate	fOSC=1.00MHz				fOSC=1.8432MHz				fOSC=2.00MHz			
	DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8K	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6K	-	-	1	8.5%	1	-25.0%	3	0.0%	1	8.5%	3	8.5%
76.8K	-	-	1	-18.6%	1	0.0%	2	0.0%	1	-18.6%	2	8.5%
115.2K	-	-	-	-	-	-	1	0.0%	-	-	1	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	-	-
Baud Rate	**fOSC=3.6864MHz**				**fOSC=4.00MHz**				**fOSC=7.3728MHz**			
	DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	-	-
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4K	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2K	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8K	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4K	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6K	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8K	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2K	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4K	-	-	1	0.0%	-	-	1	8.5%	1	0.0%	3	0.0%
250K	-	-	1	-7.8%	-	-	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	-	-	-	-	-	-	-	-	1	-7.8%
Baud Rate	**fOSC=8.00MHz**				**fOSC=11.0592MHz**				**fOSC=14.7456MHz**			
	DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1		DBLSn=0		DBLSn=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	143	0.0%	-	-	191	0.0%	-	-
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4K	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2K	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8K	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4K	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6K	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8K	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2K	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4K	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250K	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	-	-	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

15.3.11 0% error baud rate

This USART system supports the floating point counter logic for the 0% error of baud rate. By using the 8bits floating point counter logic, the cumulative error to below the decimal point can be removed.

The floating point counter value is defined by baud rate error. In the baud rate formula, USARTn_BDR is presented the integer count value. For example, If you want to use the 57600 baud rate (fXIN = 16MHz), a calculated integer count value must be 16.36 value (USARTn_BDR+1 = 16000000/(16×57600) = 17.36). Here, USARTn_BDR which can be set is the nearest big integer number 17. To realize 0% error of baud rate, floating point counter value must be 92 ((0.36) x 256 ≒ 92). Namely you have to write the 92 (decimal number) in USARTn_FPCR and 17 (decimal number) in USART_BAUD.

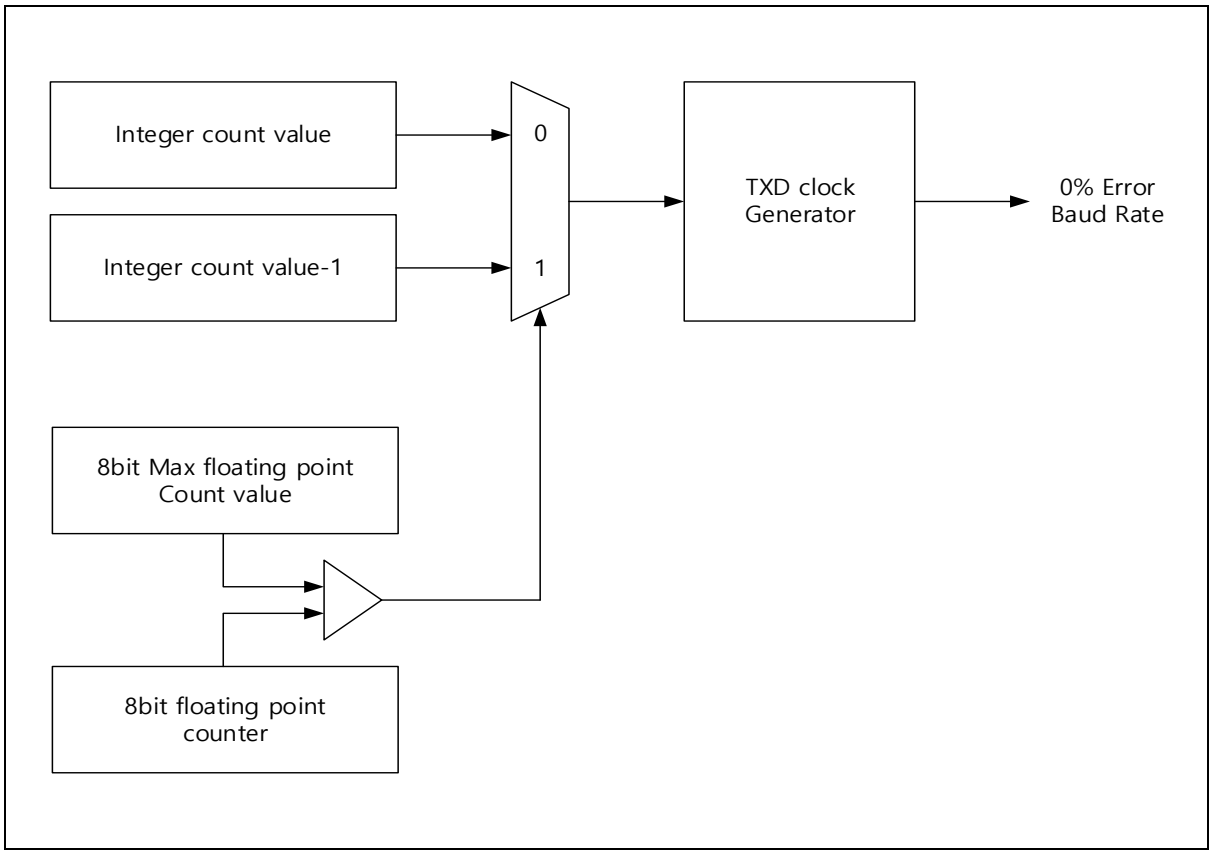


Figure 123. 0% Error Baud Rate Diagram

15.3.12 Receive time out (RTO)

This USART system supports the receive time out (RTO) interrupt. The RTO counter uses the system clock and continues counting while the RXD input is not present. If the RTO counter matches USARTn_RTO, RTOF becomes 1. RTOEN is set to 1 to enable this operation, and RTOEN is automatically set to 0 when an RTO match occurs. You can use RTO interrupts with RTOIE.

16. I2C interface

I2C is one of industrial standard serial communication protocols, and which uses 2 bus lines such as Serial Data Line (SDAn) and Serial Clock Line (SCLn) to exchange data. Because both SDAn and SCLn lines are open-drain output, each line needs pull-up resistor respectively.

I2C features the followings (n = 0 and 1):

- Compatible with I2C bus standard.
- Multi-master operation.
- Up to 1MHz data transfer read speed.
- 7-bit address.
- Support two slave address.
- Both master and slave operation.
- Bus busy detection

Table 69 introduces pins assigned for I2C interface.

Table 69. Pin Assignment of I2C: External Pins

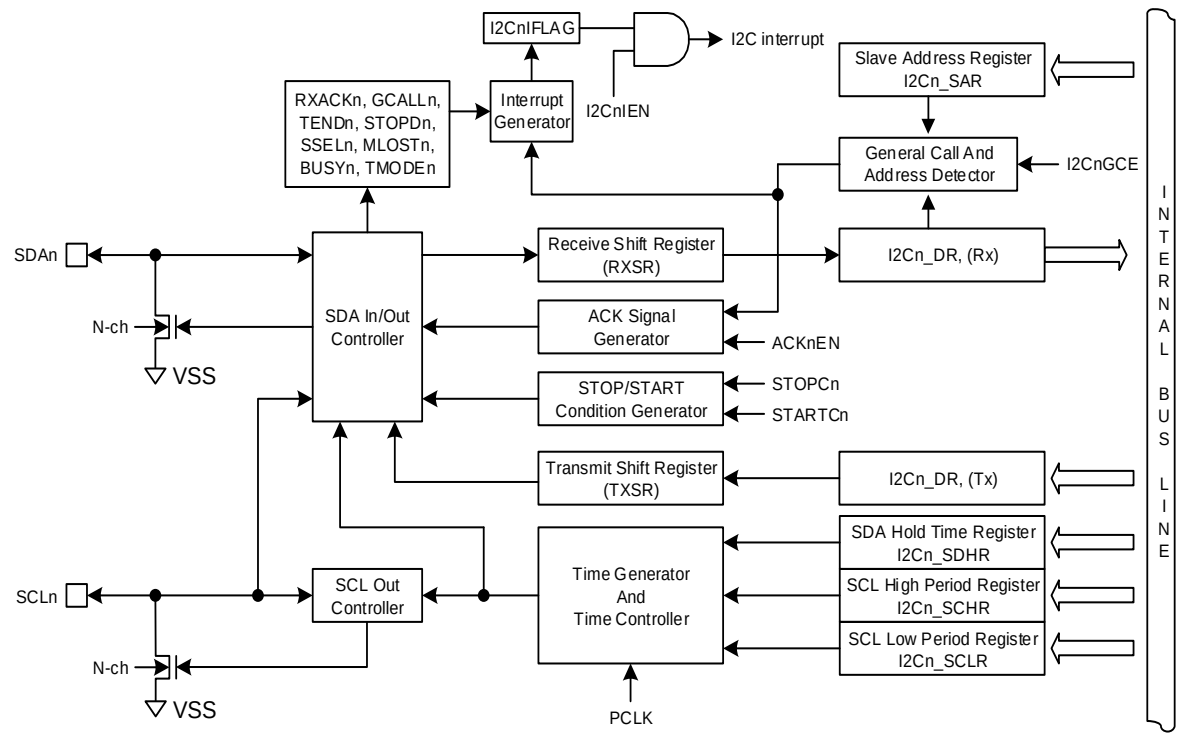
Pin name	Type	Description
SCL0	I/O	I2C channel 0 Serial clock bus line (open-drain)
SDA0	I/O	I2C channel 0 Serial data bus line (open-drain)
SCL1	I/O	I2C channel 1 Serial clock bus line (open-drain)
SDA1	I/O	I2C channel 1 Serial data bus line (open-drain)

NOTE:

1. n = 0 and 1

16.1 I2C block diagram

In this section, I2C interface block is described in a block diagram.



NOTE:

1. When the corresponding port of I2C is a sub-function for SCL/SDA pin, the SCL/SDA pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

Figure 124. I2C Block Diagram

16.2 Registers

Base address of I2C is introduced in the followings:

Table 70. Base Address of I2C Interface

Name	Base address
I2C0	0x4000_4800
I2C1	0x4000_4900

Table 71. I2C Register Map

Name	Offset	Type	Description	Reset value	Reference
I2Cn_CR	0x00	RW	I2Cn Control Register	0x0000_0000	16.2.1
I2Cn_ST	0x04	RW	I2Cn Status Register	0x0000_0000	16.2.2
I2Cn_SAR1	0x08	RW	I2Cn Slave Address Register 1	0x0000_0000	16.2.3
I2Cn_SAR2	0x0C	RW	I2Cn Slave Address Register 2	0x0000_0000	16.2.4
I2Cn_DR	0x10	RW	I2Cn Data Register	0x0000_0000	16.2.5
I2Cn_SDHR	0x14	RW	I2Cn SDA Hold Time Register	0x0000_0001	16.2.6
I2Cn_SCLR	0x18	RW	I2Cn SCL Low Period Register	0x0000_003F	16.2.7
I2Cn_SCHR	0x1C	RW	I2Cn SCL High Period Register	0x0000_003F	16.2.8
I2Cn_MR	0x20	RW	I2Cn Mode Control Register	0x0000_0000	16.2.9

NOTE:

1. n = 0 and 1

16.2.1 I2Cn_CR: I2Cn control register

The register can be set to configure I2C operation mode and simultaneously allowed for I2C transactions to be kicked off. I2Cn_CR is a 32-bit register and able to do 32/16/8-bit access.

I2C0_CR=0x4000_4800, I2C1_CR=0x4000_4900

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			I2CnIFLAG	I2CnEN	RESET	I2CnIEN	ACKnEN	IMASTERn	STOPCn	STARTCn
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RO	RW	RO	RW	RW

7	I2CnIFLAG	Interrupt flag bit	
		0	No interrupt is generated or interrupt is cleared
		1	An interrupt is generated
6	I2CnEN	Activate I2Cn Block (by supplying clock)	
		0	Disable I2C block
		1	Enable I2C block
5	RESET	Initialize internal register of I2C	
		0	No operation
		1	Initialize I2C, auto cleared
4	I2CnIEN	I2Cn Interrupt Enable bit.	
		0	Disable interrupt, operates in polling mode
		1	Enable interrupt

3	ACKnEN	Controls ACK signal generation at ninth SCL period.	
		0	No ACK signal is generated. (SDA = 1)
		1	ACK signal is generated. (SDA = 0)

NOTES:

1. ACK signal is output (SDA = 0) for the following 3 cases. Where x = 0 and 1.
2. When received address packet equals to SLAx[6:0] bits in I2Cn_SARx register.
3. When received address packet equals to value 0x00 with GCALLn enabled.
4. When I2Cn operates as a receiver (master or slave)

2	IMASTERn	Represent Operation Mode of I2Cn. This bit is cleared to "0b" on STOP condition.	
		0	I2Cn is in slave mode.
		1	I2Cn is in master mode.
1	STOPCn	STOP Condition Generation When I2Cn is master.	
		0	No effect.
		1	STOP condition is to be generated.
0	STARTCn	START Condition Generation When I2Cn is master.	
		0	No effect.
		1	START or Repeated START condition is to be generated.

16.2.2 I2Cn_ST: I2Cn status register

I2Cn_ST is a 32-bit register and able to do 32/16/8-bit access.

I2C0_ST=0x4000_4804, I2C1_ST=0x4000_4904

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			GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RO	RW

7	GCALLn	This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.	
		0	No AACK is received. (Master mode)
		1	AACK is received (Master mode). It may be set to "1b" after address transmission.
		When I2C is a slave, this bit is used to indicate general call.	
		0	General call address is not detected. (Slave mode)
		1	General call address is detected. (Slave mode)
6	TENDn	This bit is set when 1-byte of data is transferred completely.	
		0	1 byte of data is not completely transferred.
		1	1 byte of data is completely transferred.
5	STOPDn	This bit is set when a STOP condition is detected.	
		0	A STOP condition is not detected.
		1	A STOP condition is detected.
4	SSELn	This bit is set when I2C is addressed by other master.	
		0	I2C is not selected as a slave.
		1	I2C is addressed by other master and acts as a slave.

3	MLOSTn	This bit represents the result of bus arbitration in master mode.	
		0	I2C maintains bus mastership.
		1	I2C has lost bus mastership during arbitration process.
2	BUSYn	This bit reflects bus status.	
		0	I2C bus is idle, so a master can issue a START condition.
		1	I2C bus is busy.
1	TMODEn	This bit is used to indicate whether I2C is transmitter or receiver.	
		0	I2C is a receiver.
		1	I2C is a transmitter.
0	RXACKn	This bit shows the state of ACK signal.	
		0	No ACK is received.
		1	ACK is received at ninth SCL period.

NOTES:

1. The GCALLn, TENDn, STOPDn, SSELn, and MLOSTn bits can be source of interrupt.
2. When an I2C interrupt occurs except for deep sleep mode, the SCL line is held low.
3. To release SCL, Clear to "0b" all interrupt source bits in I2Cn_ST register.
4. The GCALLn, TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared when "1b" is written to the corresponding bit.

16.2.3 I2Cn_SAR1: I2Cn slave address register 1

I2Cn_SAR1 is a 32-bit register and able to do 32/16/8-bit access.

I2C0_SAR1=0x4000_4808, I2C1_SAR1=0x4000_4908

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			SLAn	GCALLnEN
–			0000000	0
-			RW	RW

7 1	SLAn	These bits configure the slave address 0 in slave mode.
0	GCALLnEN	This bit decides whether I2Cn allows general call address 0 or not in I2Cn slave mode.
	0	Ignore general call address 0.
	1	Allow general call address 0.

16.2.4 I2Cn_SAR2: I2Cn slave address register 2

I2Cn_SAR2 is a 32-bit register and able to do 32/16/8-bit access.

I2C0_SAR2=0x4000_480C, I2C1_SAR2=0x4000_490C

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			SLAn	GCALLnEN
0x000000			0000000	0
-			RW	RW

7 1	SLAn	These bits configure the slave address 1 in slave mode.	
0	GCALLnEN	This bit decides whether I2Cn allows general call address 1 or not in I2Cn slave mode.	
		0	Ignore general call address 1.
		1	Allow general call address 1.

16.2.5 I2Cn_DR: I2Cn data register

I2Cn_DR is a 32-bit register and able to do 32/16/8-bit access.

I2C0_DR=0x4000_4810, I2C1_DR=0x4000_4910

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			DATA
			0x0F
			RW

7 0	DATA	The I2Cn_DR Transmit buffer and Receive buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the I2Cn_DR register. Reading the I2Cn_DR register returns the contents of the Receive Buffer.

16.2.6 I2Cn_SDAHR: I2Cn SDA hold time register

I2Cn_ SDAHR is a 32-bit register and able to do 32/16/8-bit access.

I2C0_SDHR=0x4000_4814, I2C1_SDHR=0x4000_4914

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			HLDT
			0x0F
			RW

7 0	HLDT	This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after tPCLK X (I2Cn_SDHR+2). In master mode, load half the value of I2Cn_SCLR to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after tPCLK X (I2Cn_SDHR+2) in master mode. So, to insure operation in slave mode, the value tPCLK X (I2Cn_SDHR + 2) must be smaller than the period of SCL.

16.2.7 I2Cn_SCLLR: I2Cn SCL low period register

I2Cn_ SCLLR is a 32-bit register and able to do 32/16/8-bit access.

I2C0_SCLR=0x4000_4818, I2C1_SCLR=0x4000_4918

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			SCLL
			0xF
			RW

7 0	SCLL	This register defines the low period of SCL in master mode. The base clock is PCLK and the period is calculated by the formula: tPCLK X (4 X I2Cn_SCLR + 2) where tPCLK is the period of PCLK.

16.2.8 I2Cn_SCLHR: I2Cn SCL high period register

I2Cn_ SCLHR is a 32-bit register and able to do 32/16/8-bit access.

I2C0_SCHR=0x4000_481C, I2C1_SCHR=0x4000_491C

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			SCLH
			0xF
			RW

7 0	SCLH	This register defines the high period of SCL in master mode. The base clock is PCLK and the period is calculated by the formula: tPCLK X (4 X I2Cn_SCHR + 2) where tPCLK is the period of PCLK.

16.2.9 I2Cn_MR: I2Cn mode control register

I2C0_SCHR=0x4000_4820, I2C1_SCHR=0x4000_4920

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				DIS_SDAH
				0
				RW

0	DIS_SDAH	Disable SDA hold time	
		0	Enable SDA hold time
			When I2CST is written, the MSB of the transmitted data of I2CDR is on the output to SDA port before three system clocks from the rising edge of SCL. The other bits after the MSB bit are outputted to SDA port in the timing Controlled by I2CSDAHR from the falling edge of SCL.
		1	Disable SDA hold time
			When I2CDR is written, the MSB of the transmitted data of I2CDR is on the output to SDA port directly. The other bits after the MSB bit are outputted after the falling edge of SCL irrespective of I2CSDAHR.

16.3 Functional description

16.3.1 I2C bit transfer

Data on the SDAn line must be stable during HIGH period of the clock, SCLn. HIGH or LOW state of the data line can only change when the clock signal on the SCLn line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

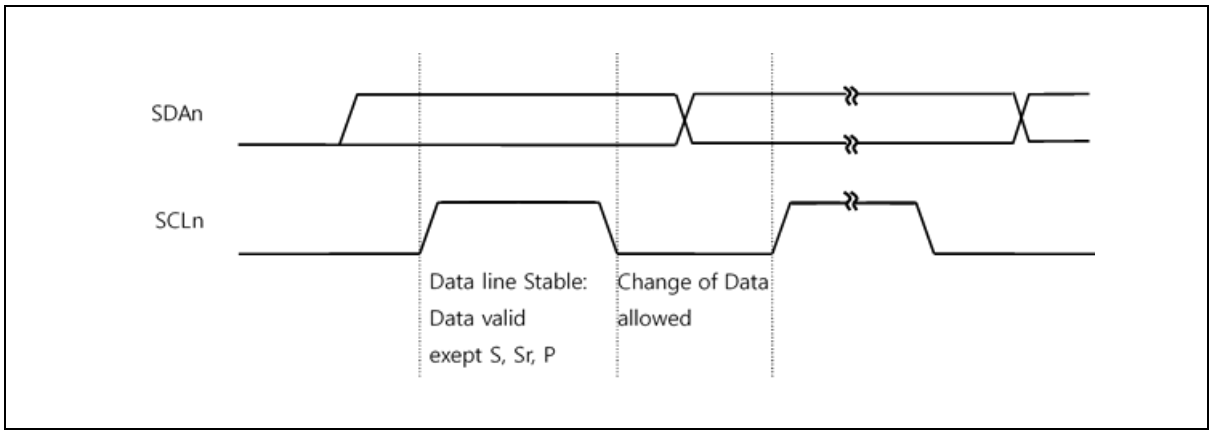


Figure 125. I2C Bus Bit Transfer (n = 0 and 1)

16.3.2 START/repeated START/STOP

One master can issue a START (S) condition to notice other devices connected to the SCLn, SDAn lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

- A high to low transition on the SDAn line while SCLn is high defines a START (S) condition.
- A low to high transition on the SDAn line while SCLn is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, ie, the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

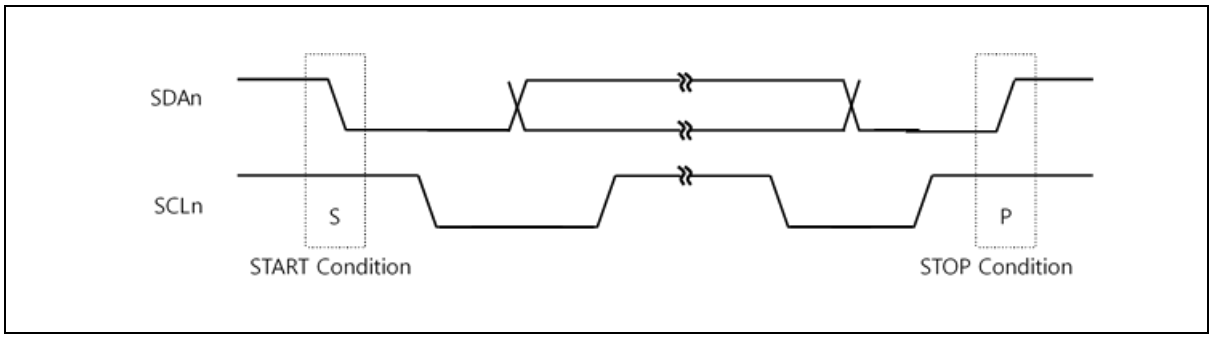


Figure 126. START and STOP Condition (n = 0 and 1)

16.3.3 Data transfer

Every byte put on the SDAn line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first.

If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCLn LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCLn.

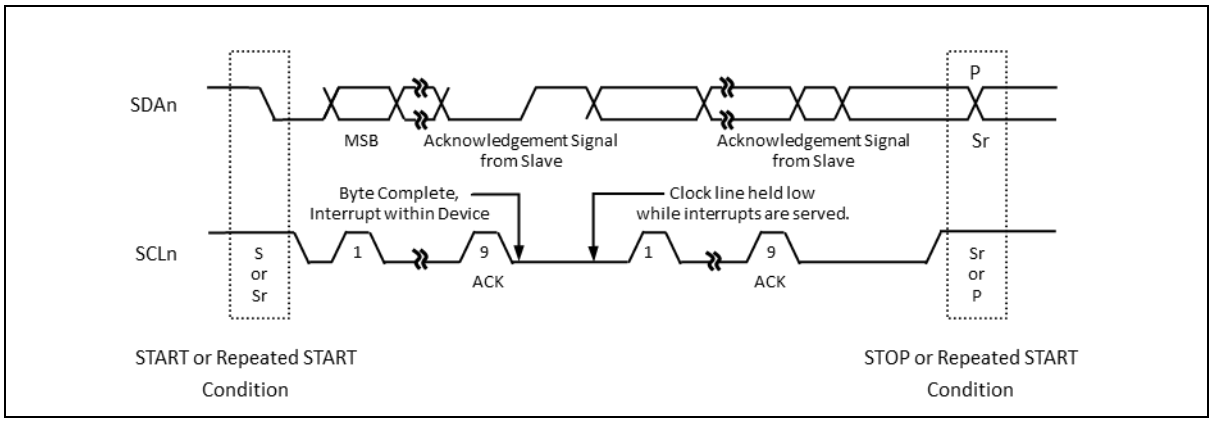


Figure 127. I2C Bus Data Transfer (n = 0 and 1)

16.3.4 Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDAn line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDAn line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it's performing some real time function, the data line must be left HIGH by the slave.

And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDAn line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating any acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

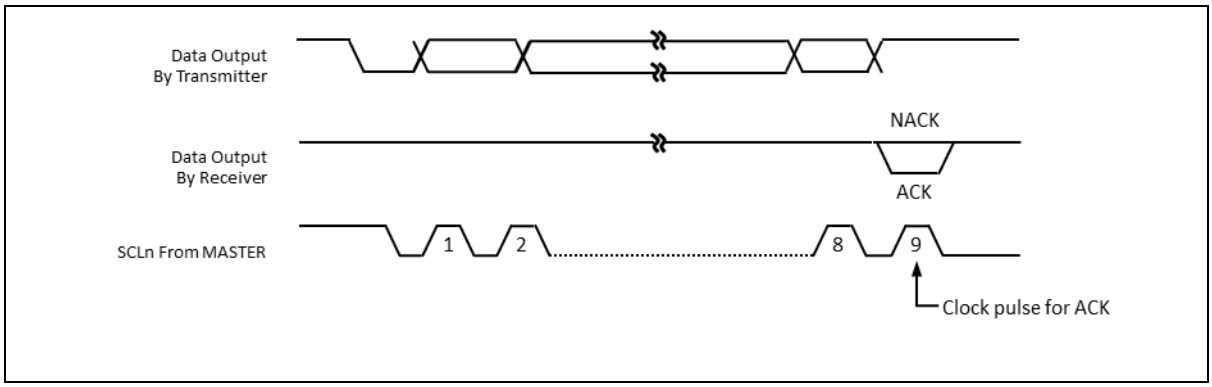


Figure 128. I2C Bus Acknowledge (n = 0 and 1)

16.3.5 Synchronization/Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCLn line. This means that a HIGH to LOW transition on the SCLn line will cause the devices concerned to start counting off their LOW period and it will hold the SCLn line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCLn line if another clock is still within its LOW period. In this way, a synchronized SCLn clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDAn line, while the SCLn line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

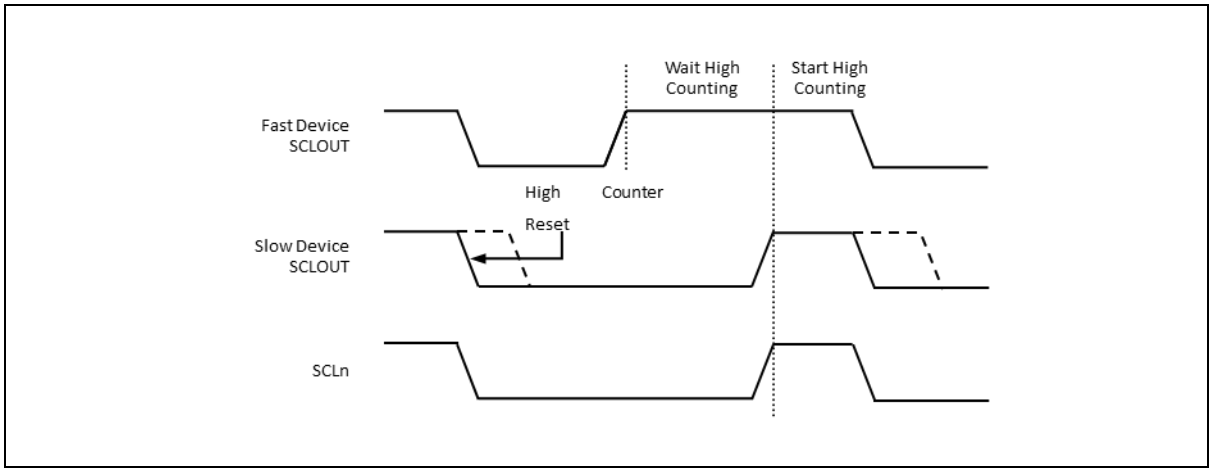


Figure 129. Clock Synchronization during the Arbitration Procedure (n = 0 and 1)

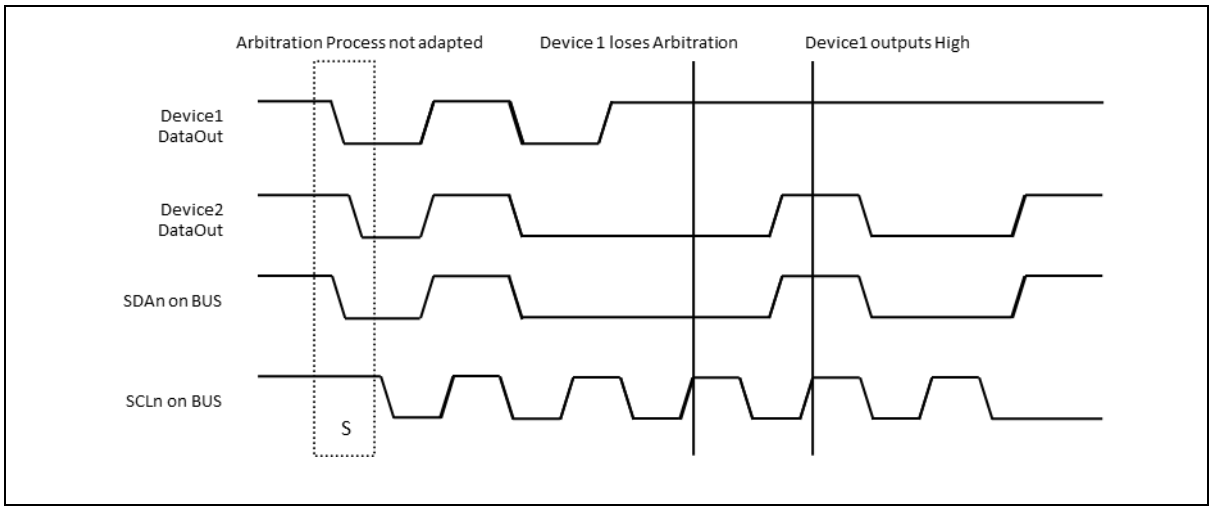


Figure 130. Arbitration Procedure between Two Masters (n = 0 and 1)

16.3.6 I2C operation

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during an I2C byte transfer.

Note that when an I2C interrupt is generated, I2CnIFLAG flag in I2CnIEN register is set, it is cleared when all interrupt source bits in the I2Cn_ST register are cleared to "0b". When I2C interrupt occurs, the SCLn line is hold LOW until clearing "0b" all interrupt source bits in I2Cn_ST register. When the I2CnIFLAG flag is set, the I2Cn_ST contains a value indicating the current state of the I2C bus. According to the value in I2Cn_ST, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below. (n = 0 and 1)

Master transmitter

To operate I2C in master transmitter, follow the recommended steps below:

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
2. Load SLA+W into the I2Cn_DR where SLA is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that I2Cn_DR is used for both address and data.
3. Configure baud rate by writing desired value to both I2Cn_SCLR and I2Cn_SCHR for the Low and High period of SCLn line.
4. Configure the I2Cn_SDHR to decide when SDAn changes value from falling edge of SCLn. If SDA should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in I2Cn_DR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in I2Cn_ST is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in I2Cn_ST is set, the ACKnEN bit in I2Cn_CR must be set and the received 7-bit address must equal to the SLAn bits in I2Cn_SAR1/2.
   In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (master) can choose one of the following cases regardless of the reception of ACK signal from slave.

   Case 1. Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

   Case 2. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2Cn_CR.

   Case 3. Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line.

   For the Case 1, move to step 7. For the Case 2, move to step 9 to handle STOP interrupt. For the Case 3, move back to step 6 after transmitting the data in I2Cn_DR and if transfer direction bit is '1' go to master receiver section.

7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.

8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOSTn bit in I2Cn_ST is set. If then, I2C waits in idle state. When the data in I2Cn_DR is transmitted completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   Case A. Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

   Case B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in I2CnCR.

   Case C. Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set the STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCL line. For the Case A, move back to step 7. For the Case B, move to step 9 to handle STOP interrupt. For the Case C, move back to step 6 after transmitting the data in I2C_DR, and if transfer direction bit is '1' go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0" to I2CnST. After this, I2C enters in idle state.

Master receiver

To operate I2C in master receiver, follow the recommended steps below:

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
2. Load SLA+R into the I2Cn_DR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that I2Cn_DR is used for both address and data.
3. Configure baud rate by writing desired value to both I2Cn_SCLR and I2Cn_SCHR for the Low and High period of SCLn line.
4. Configure the I2CnSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in I2Cn_DR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in I2Cn_ST is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in I2Cn_ST is set, the ACKnEN bit in I2Cn_CR must be set and the received 7-bit address must equal to the SLAn bits in I2Cn_SAR1/2. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (master) can choose one of the following cases according to the reception of ACK signal from slave.

   Case 1. Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKnEN bit in I2Cn_CR to decide whether I2C Acknowledges the next data to be received or not.

   Case 2. Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPCn bit in I2Cn_CR.

   Case 3. Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case 1, move to step 7. For the Case 2, move to step 9 to handle STOP interrupt. For the Case 3, move to step 6 after transmitting the data in I2Cn_DR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in I2Cn_ST.

   Case A. Master continues to receive data from slave. To do this, set ACKnEN bit in I2Cn_CR to acknowledge the next data to be received.

   Case B. Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKnEN bit in I2Cn_CR.

   Case C. Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPCn bit in I2Cn_CR.

   Case D. No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CnDR and set the STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case A and B, move to step 7. For the Case C, move to step 9 to handle STOP interrupt. For the Case D, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0" value to I2Cn_ST. After this, I2C enters idle state.

Slave transmitter

To operate I2C in slave transmitter, follow the recommended steps below:

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into I2Cn_SDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CnSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit and I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLAn bits in I2Cn_SAR1/2. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in I2Cn_SAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to I2Cn_DR and clear to "0b" all interrupt source bits in I2Cn_ST to release SCLn line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases:

   Case 1. No ACK signal is detected and I2C waits STOP or repeated START condition.

   Case 2. ACK signal from master is detected. Load data to transmit into I2Cn_DR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case 1, move to step 7 to terminate communication. For the Case 2, move back to step 5. In either case, a repeated START condition can be detected. For that case, move back to step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0" to I2Cn_ST. After this, I2C enters idle state.

Slave receiver

To operate I2C in slave receiver, follow the recommended steps below:

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into I2Cn_SDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2Cn_SDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit in I2CnCR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2C_SAR. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in I2Cn_SAR1/2, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLA bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, clear to "0b" all interrupt source bits in I2Cn_ST to release SCLn line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases:

   Case 1. No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.

   Case 2. ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, clear all interrupt source bits in I2CnST to "0b" to release SCLn line. For the Case 1, move to step 7 to terminate communication. For the Case 2, move back to step 5. In either case, a repeated START condition can be detected. For that case, move back to step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear I2CnST, write "0" to I2CnST. After this, I2C enters idle state.

17. Serial peripheral interface (SPI)

A channel serial interface is provided for synchronous serial communications with external peripherals. SPI block support both of master and slave mode. Four signals will be used for SPI communication such as SS, SCK, MOSI, and MISO.

SPI of A31G32x series features the followings:

- Master or Slave operation.
- Programmable clock polarity and phase.
- 8, 9, 16, 17-bit wide transmit/receive register.
- 8, 9, 16, 17-bit wide data frame.
- Loop-back mode.
- Programmable start, burst, and stop delay time.
- DMA transfer operation.

Table 72 introduces pins assigned for SPI.

Table 72. Pin Assignment of SPI: External Pins

Pin name	Type	Description
SSn	I/O	SPIn Slave select input / output
SCKn	I/O	SPIn Serial clock input / output
MOSIn	I/O	SPIn Serial data (Master output, Slave input)
MISOn	I/O	SPIn Serial data (Master input, Slave output)

NOTE: n = 20 and 21

17.1 SPI block diagram

In this section, SPI is described in a block diagram in Figure 131.

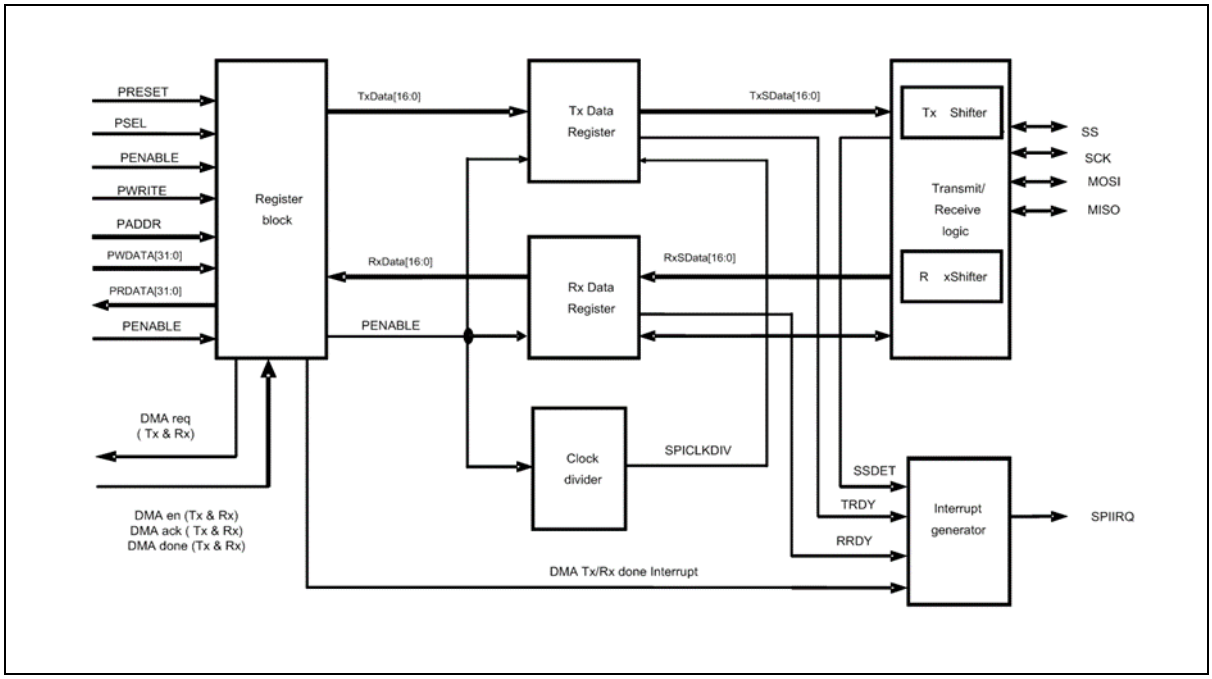


Figure 131. SPI Block Diagram

17.2 Registers

Base address of SPI is introduced in the followings:

Table 73. Base Address of SPI

Name	Base address
SPI20	0x4000_4C00
SPI21	0x4000_4D00

Table 74. SPI Register Map

Name	Offset	Type	Description	Reset value	Reference
SPIn.TDR	0x00	W	SPIn Transmit Data Register	0x000000	17.2.1
SPIn.RDR	0x00	R	SPIn Receive Data Register	0x000000	17.2.2
SPIn.CR	0x04	RW	SPIn Control Register	0x001020	17.2.3
SPIn.SR	0x08	RW	SPIn Status Register	0x000006	17.2.4
SPIn.BR	0x0C	RW	SPIn Baud rate Register	0x0000FF	17.2.5
SPIn.EN	0x10	RW	SPIn Enable register	0x000000	17.2.6
SPIn.LR	0x14	RW	SPIn delay Length Register	0x010101	17.2.7

17.2.1 SPIn.TDR: SPI transmit data register

SPIn.TDR is a 17-bit sized read/write register. It contains serial transmit data.

SPI20.TDR=0x4000_4C00, SPI21.TDR=0x4000_4D00

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		TDR	
0		0x00000	
-		RW	

16 0	TDR	Transmit Data Register

17.2.2 SPIn.RDR: SPI receive data register

SPIn.RDR is a 17-bit sized read/write register. It contains serial receive data.

SPI20.RDR=0x4000_4C00, SPI21.RDR=0x4000_4D00

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		RDR	
0		0x00000	
-		RW	

16 0	RDR	Receive Data Register

17.2.3 SPIn.CR: SPI control register

SPIn.CR is a 20-bit sized read/write register and can be set to configure SPI operation mode.

SPI20.CR=0x4000_4C04, SPI21.CR=0x4000_4D04

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		TXBC	RXBC	DTXIE	DRXIE	SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMARK	SS0MO	SS0POL	Reserved	MS	MSBF	CPHA	CPOL	BITSZ
0		0	0	0	0	0	0	0	0	1	0	0	0	0	-	1	0	0	0	00
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW

20	TXBC	Tx buffer clear bit.	
		0	No action
		1	Clear Tx buffer
19	RXBC	Rx buffer clear bit	
		0	No action
		1	Clear Rx buffer
18	TXDIE	DMA Tx Done Interrupt Enable bit.	
		0	DMA Tx Done Interrupt is disabled.
		1	DMA Tx Done Interrupt is enabled.
17	RXDIE	DMA Rx Done Interrupt Enable bit.	
		0	DMA Rx Done Interrupt is disabled.
		1	DMA Rx Done Interrupt is enabled.
16	SSCIE	SS Edge Change Interrupt Enable bit.	
		0	nSS interrupt is disabled.
		1	nSS interrupt is enabled for both edges (L→H, H→L)
15	TXIE	Transmit Interrupt Enable bit.	
		0	Transmit Interrupt is disabled.
		1	Transmit Interrupt is enabled.
14	RXIE	Receive Interrupt Enable bit.	
		0	Receive Interrupt is disabled.
		1	Receive Interrupt is enabled.
13	SSMOD	SS Auto/Manual output select bit.	
		0	SS output is not set by SSOUT (SPInCR[12]). SS signal is in normal operation mode.
		1	SS output signal is set by SSOUT.
12	SSOUT	SS output signal select bit.	
		0	SS output is 'L.'
		1	SS output is 'H'.
11	LBE	Loop-back mode select bit in master mode.	
		0	Loop-back mode is disabled.
		1	Loop-back mode is enabled.
10	SSMASK	SS signal masking bit in slave mode.	
		0	SS signal masking is disabled. Receive data when SS signal is active.
		1	SS signal masking is enabled. Receive data at SCLK edges. SS signal is ignored.
9	SSMO	SS output signal select bit.	
		0	SS output signal is disabled.
		1	SS output signal is enabled.
8	SSPOL	SS signal Polarity select bit.	
		0	SS signal is Active-Low.
		1	SS signal is Active-High.
5	MS	Master/Slave select bit.	
		0	SPI is in Slave mode.
		1	SPI is in Master mode.
4	MSBF	MSB/LSB Transmit select bit.	
		0	LSB is transferred first.
		1	MSB is transferred first.
3	CPHA	SPI Clock Phase bit.	
		0	Sampling of data occurs at odd edges (1, 3, 5,…, 15).
		1	Sampling of data occurs at even edges (2, 4, 6,…, 16).

2	CPOL	SPI Clock Polarity bit.	
		0	Active-high clocks selected.
		1	Active-low clocks selected.
1 0	BITSZ	Transmit/Receive Data Bits select bit.	
		00	8 bits
		01	9 bits
		10	16 bits
		11	17 bits

NOTES:

1. CPOL=0, CPHA=0 : data sampling at rising edge, data changing at falling edge
2. CPOL=0, CPHA=1 : data sampling at falling edge, data changing at rising edge
3. CPOL=1, CPHA=0 : data sampling at falling edge, data changing at rising edge
4. CPOL=1, CPHA=1 : data sampling at rising edge, data changing at falling edge

17.2.4 SPIn.SR: SPI status register

SPIn.SR is a 10-bit sized read/write register. It contains the status of SPI interface.

SPI20.SR=0x4000_4C08, SPI21.SR=0x4000_4D08

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	TXDMAF	RXDMAF	Reserved	SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
-	0	0	-	0	0	0	0	1	1	0
-	RC1	RC1	-	RC1	RC1	RC1	RC1	R	R	R

9	TXDMAF	DMA Transmit Operation Complete flag. (DMA to SPI)	
		0	DMA Transmit Op is working or is disabled.
		1	DMA Transmit Op is done.
8	RXDMAF	DMA Receive Operation Complete flag. (SPI to DMA)	
		0	DMA Receive Operation is working or is disabled.
		1	DMA Transmit Op is done.
6	SSDET	The rising or falling edge of SS signal Detect flag.	
		0	SS edge is not detected.
		1	SS edge is detected. The bit is cleared when it is written as "0".
5	SSON	SS signal Status flag.	
		0	SS signal is inactive.
		1	SS signal is active.
4	OVRF	Receive Overrun Error flag.	
		0	Receive Overrun error is not detected.
		1	Receive Overrun error is detected. This bit is cleared by writing or reading SPInRDR.

3	UDRF	Transmit Underrun Error flag.	
		0	Transmit Underrun is not occurred.
		1	Transmit Underrun is occurred. This bit is cleared by writing or reading SPInTDR.
2	TXIDLE	Transmit/Receive Operation flag.	
		0	SPI is transmitting data
		1	SPI is in IDLE state.
1	TRDY	Transmit buffer Empty flag.	
		0	Transmit buffer is busy.
		1	Transmit buffer is ready. This bit is cleared by writing data to SPInTDR.
0	RRDY	Receive buffer Ready flag.	
		0	Receive buffer has no data.
		1	Receive buffer has data. This bit is cleared by writing data to SPInRDR.

17.2.5 SPIn.BR: SPI baud rate register

SPIn.BR is a 16-bit sized read/write register. Baud rate can be set by writing the register.

SPI20.BR=0x4000_4C0C, SPI21.BR=0x4000_4D0C

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	BR	
-	0x00FF	
-	RW	

15 0	BR	Baud rate setting bits Baud Rate = PCLK / (BR + 1)

NOTES:

1. BR[15:0] must be set 2 or greater. (BR[15:0] ≥ 2)
2. For SPI speed, it is recommended to set the BR value to 2 or higher so that the SPI input clock is divided by at least 3.

   e.g., PCLK = 24 MHz, BR = 2, SPI Freq. = 24 MHz / (2 + 1) = 8 MHz

17.2.6 SPIn.EN: SPI enable register

SPIn.EN is a bit sized read/write register. It contains SPI enable bit.

SPI20.EN=0x4000_4C10, SPI21.EN=0x4000_4D10

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	ENABLE
-	0
-	RW

0	ENABLE	SPI Enable bit
		SPI is disabled. SPInSR is initialized by writing "0" to this bit but other registers aren't initialized.
		SPI is enabled. When this bit is written as "1", the dummy data of transmit buffer will be shifted. To prevent this, write data to SPTDR before this bit is active.

17.2.7 SPIn.LR: SPI delay length register

SPIn.LR is a 24-bit sized read/write register. It contains start, burst, and stop length value.

SPI20.LR=0x4000_4C14, SPI21.LR=0x4000_4D14

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	SPL	BTL	STL
-	0x01	0x01	0x01
-	RW	RW	RW

23 16	SPL	StoPLength value
		0x01 to 0xFF: 1 to 255 SCLKs. (SPL ≥ 1)
15 8	BTL	BursTLength value
		0x01 to 0xFF: 1 to 255 SCLKs. (BTL ≥ 1)
7 0	STL	STart Length value
		0x01 to 0xFF: 1 to 255 SCLKs. (STL ≥ 1)

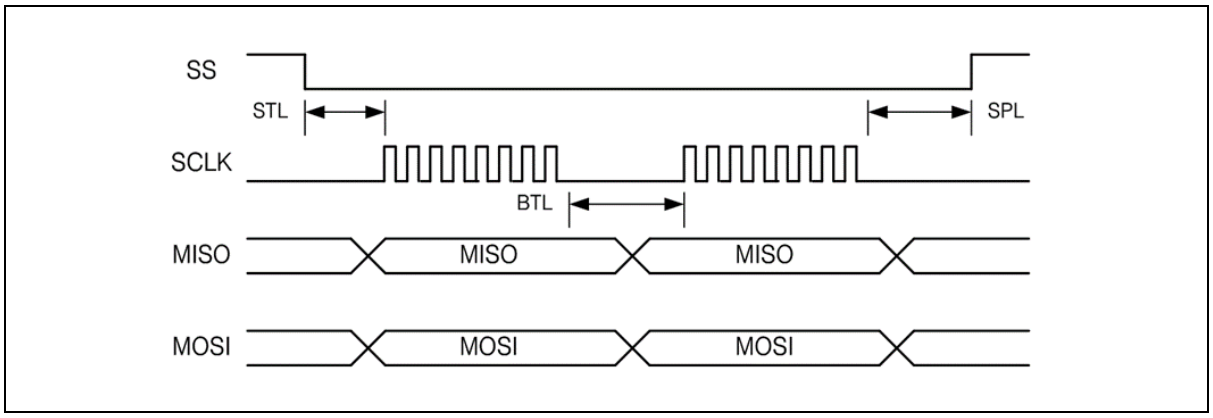


Figure 132. SPI wave form (STL, BTL and SPL)

17.3 Functional description

SPI Transmit block and Receive block share Clock Gen Block but they are independent each other. Transmit block and Receive block have double buffers and SPI is available for back to back transfer operation.

17.3.1 SPI timing

SPI has four modes of operation. These modes essentially control the way data is clocked in or out of an SPI device. The configuration is done by two bits in the SPI control register (SPInCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock. The clock phase (CPHA) control bit selects one of the two fundamentally different transfer formats.

To ensure a proper communication between master and slave both devices have to run in the same mode. This can require a reconfiguration of the master to match the requirements of different peripheral slaves.

The clock polarity has no significant effect on the transfer format. Switching this bit causes the clock signal to be inverted (active high becomes active low and idle low becomes idle high). The settings of the clock phase, however, selects one of the two different transfer timings, which are described closer in the next two chapters. Since the MOSI and MISO lines of the master and the slave are directly connected to each other, the diagrams show the timing of both device, master and slave.

The nSS line is the slave select input of the slave. The nSS pin of the master is not shown in the diagrams. It has to be inactive by a high level on this pin (if configured as input pin) or by configuring it as an output pin.

The timing of a SPI transfer where CPHA is zero is shown in Figure 133 and Figure 134. Two wave forms are shown for the SCK signal -one for CPOL equals zero and another for CPOL equals one.

When the SPI is configured as a slave, the transmission starts with the falling edge of the /SS line. This activates the SPI of the slave and the MSB of the byte stored in its data register (SPInTDR) is output on the MISO line. The actual transfer is started by a software write to the SPInTDR of the master. This causes the clock signal to be generated. In cases where the CPHA equals zero, the SCLK signal remains zero for the first half of the first SCLK cycle. This ensures that the data is stable on the input lines of both the master and the slave.

The data on the input lines is read with the edge of the SCLK line from its inactive to its active. The edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one) causes the data to be shifted one bit further so that the next bit is output on the MOSI and MISO lines.

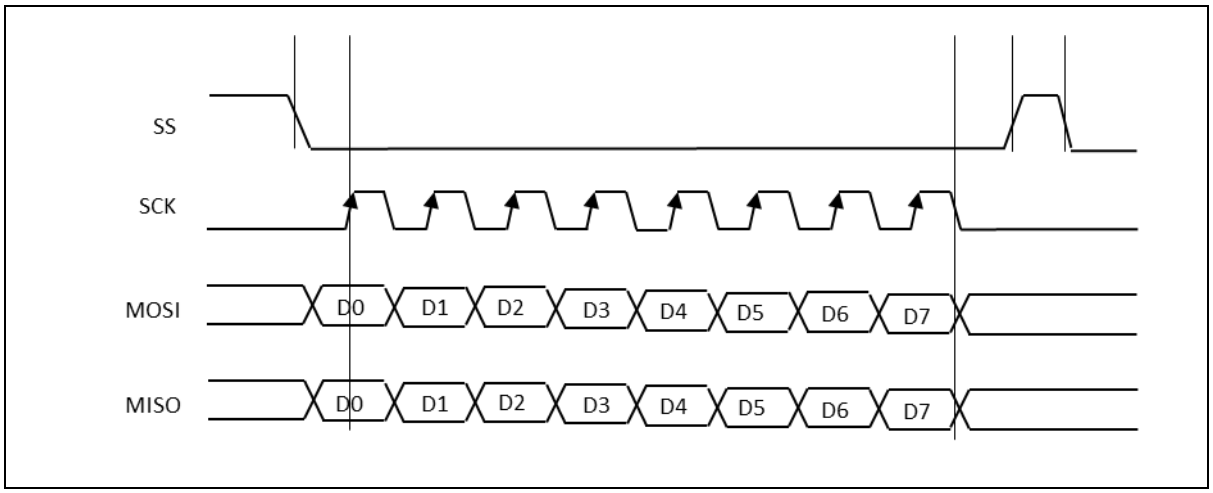


Figure 133. SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

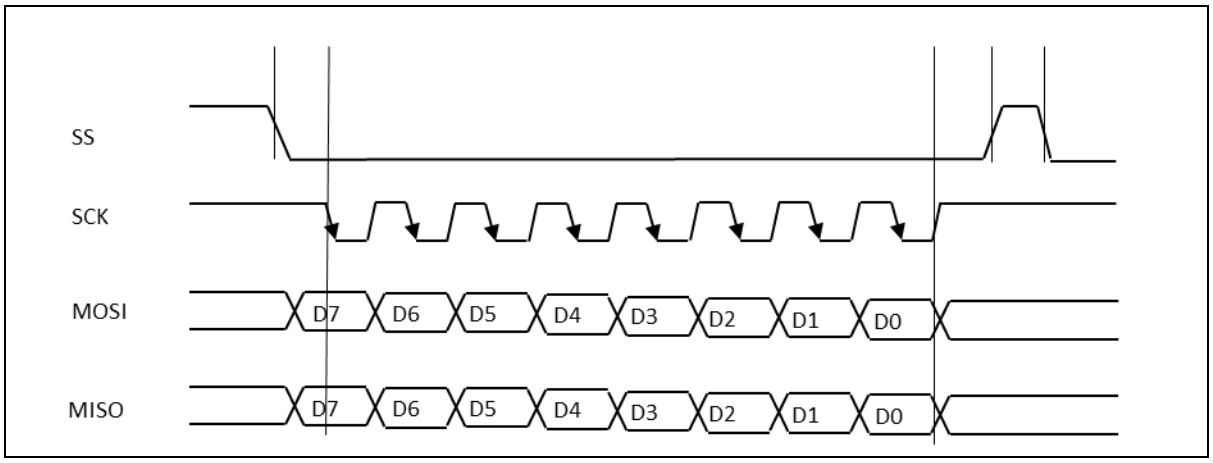


Figure 134. SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of a SPI transfer where CPHA is one is shown in Figure 135 and Figure 136. Two wave forms are shown for the SCLK signal -one for CPOL equals zero and another for CPOL equals one.

Like in the previous cases the falling edge of the nSS lines selects and activates the slave. Compared to the previous cases, where CPHA equals zero, the transmission is not started and the MSB is not output by the slave at this stage. The actual transfer is started by a software write to the SPInTDR of the master what causes the clock signal to be generated. The first edge of the SCLK signal from its inactive to its active state (rising edge if CPOL equals zero and falling edge if CPOL equals one) causes both the master and the slave to output the MSB of the byte in the SPInTDR.

As shown in Figure 133 and Figure 134, there is no delay of half a SCLK-cycle. The SCLK line changes its level immediately at the beginning of the first SCLK-cycle. The data on the input lines is read with the edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one). After eight clock pulses the transmission is completed.

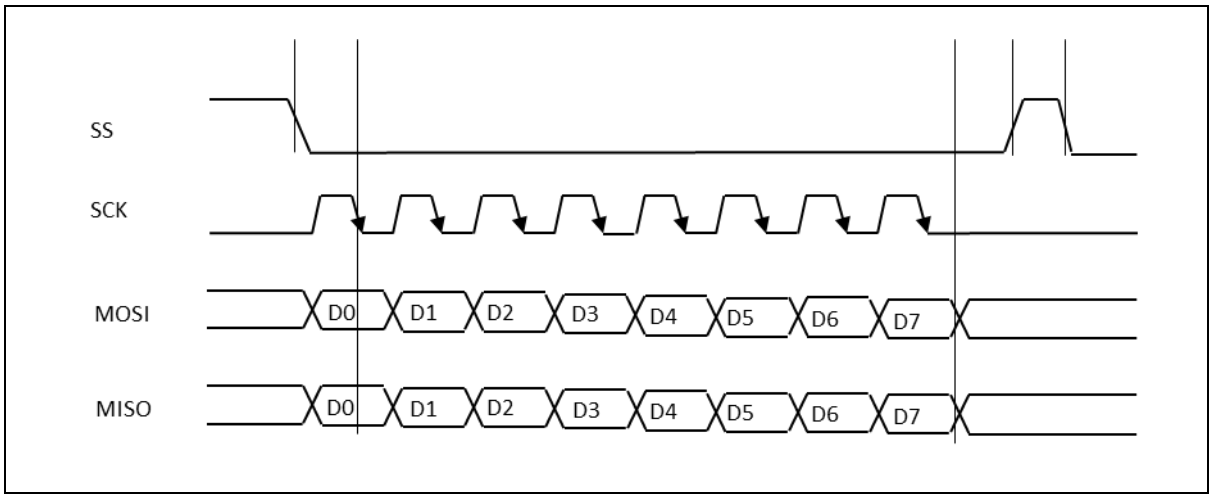


Figure 135. SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

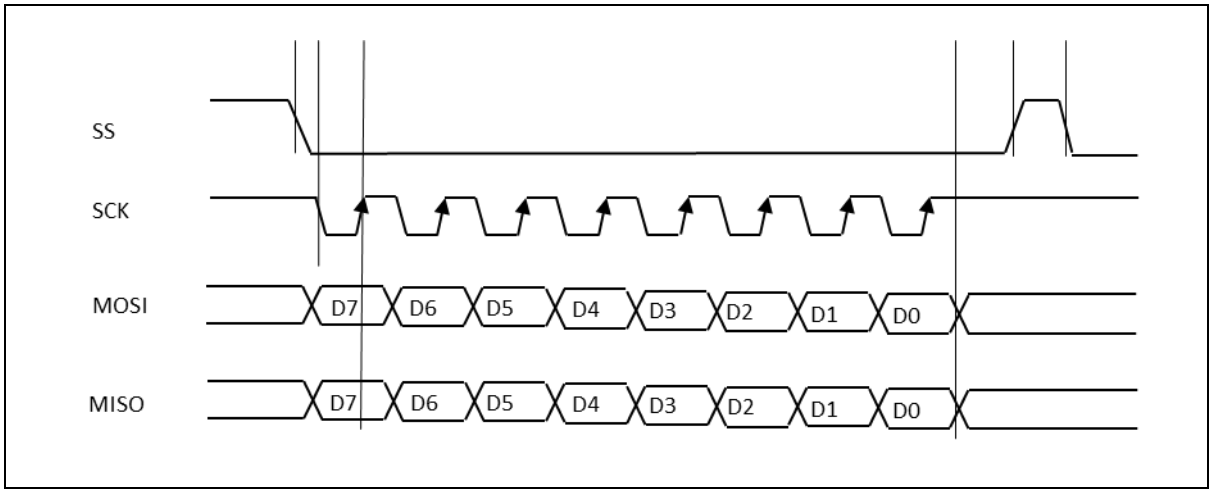


Figure 136. SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

17.3.2 DMA handshake

SPI supports DMA handshaking operation. In order to operate DMA handshake, DMA registers should be set first (see chapter 7. DMAC). As Transmitter and Receiver are independent each other, SPI can operate the two channels at the same time.

After DMA channel for receiver is enabled and receive buffer is filled, SPI sends Rx request to DMA to empty the buffer and waits ACK signal from DMA. If Receive buffer is filled again after ACK signal, SPI sends Rx request. If DMA Rx DONE becomes high, RXDMAF (SPInSR[8]) goes "1" and an interrupt is serviced when RXDIE (SPInCR[17]) is set.

Likewise, if transmit buffer is empty after DMA channel for transmitter is enabled, SPI sends Tx request to DMA to fill the buffer and waits ACK signal from DMA. If transmit buffer is empty again after ACK signal, SPI sends Tx request. If DMA Tx DONE becomes high, TXDMAF (SPInSR[9]) goes "1" and an interrupt is serviced when TXDIE (SPInCR[18]) is set.

Slave transmitter sends dummy data at the first transfer (8 to 17 SCLKs) in DMA handshake mode.

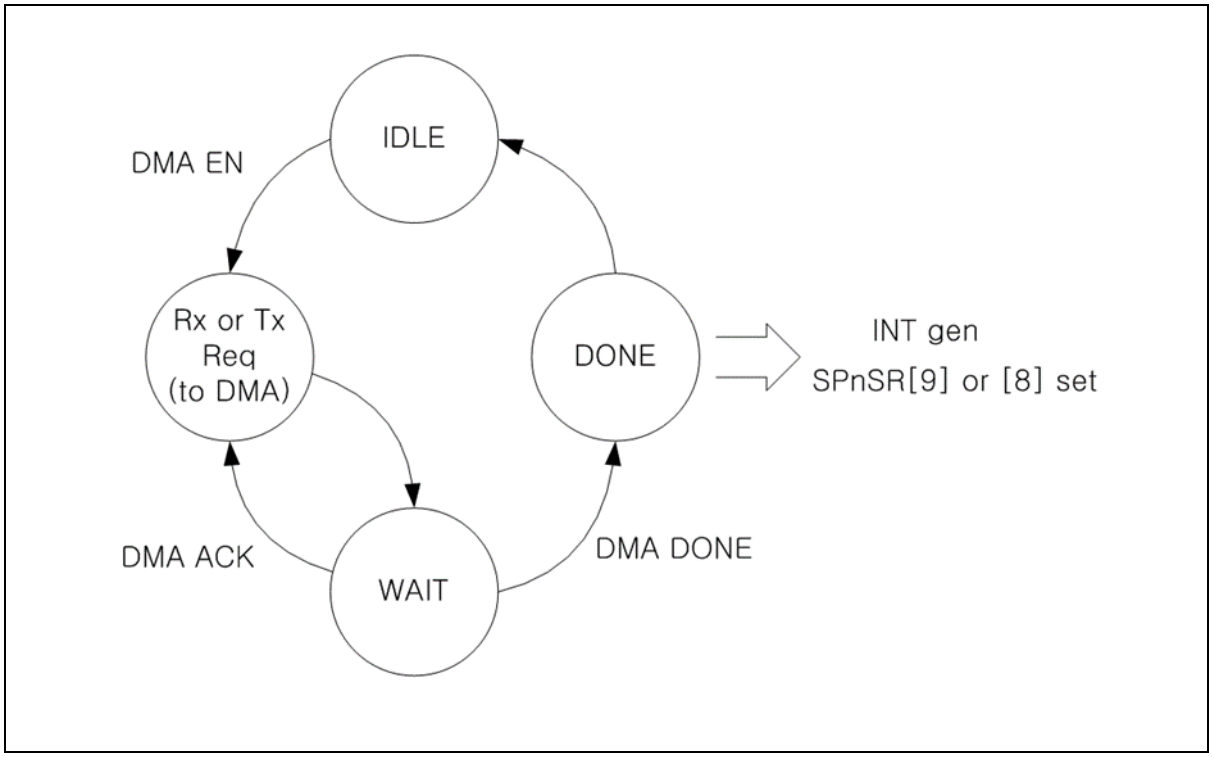


Figure 137. DMA Handshake Flow Chart

18. 12-bit ADC

ADC block of A31G32x series consists of an independent ADC unit featuring the followings:

- 16 Channel Analog Input
- Single mode and Continuous conversion mode
- Maximum 8 sequential conversion support
- Software trigger support
- Three internal trigger source (PWM, TIMER) Support
- Adjustable sample and hold time

Table 75 introduces pins assigned for ADC.

Table 75. Pin Assignment of ADC: External Signal

Pin name	Type	Description
AVDD	P	Analog Power(3.0V to VDD)
AVSS	P	Analog GND
AN0	A	ADC Input 0
AN1	A	ADC Input 1
AN2	A	ADC Input 2
AN3	A	ADC Input 3
AN4	A	ADC Input 4
AN5	A	ADC Input 5
AN6	A	ADC Input 6
AN7	A	ADC Input 7
AN8	A	ADC Input 8
AN9	A	ADC Input 9
AN10	A	ADC Input 10
AN11	A	ADC Input 11
AN12	A	ADC Input 12
AN13	A	ADC Input 13
AN14	A	ADC Input 14
AN15	A	ADC Input 15

18.1 12-bit ADC block diagram

In this section, 12-bit ADC is described in a block diagram in Figure 138.

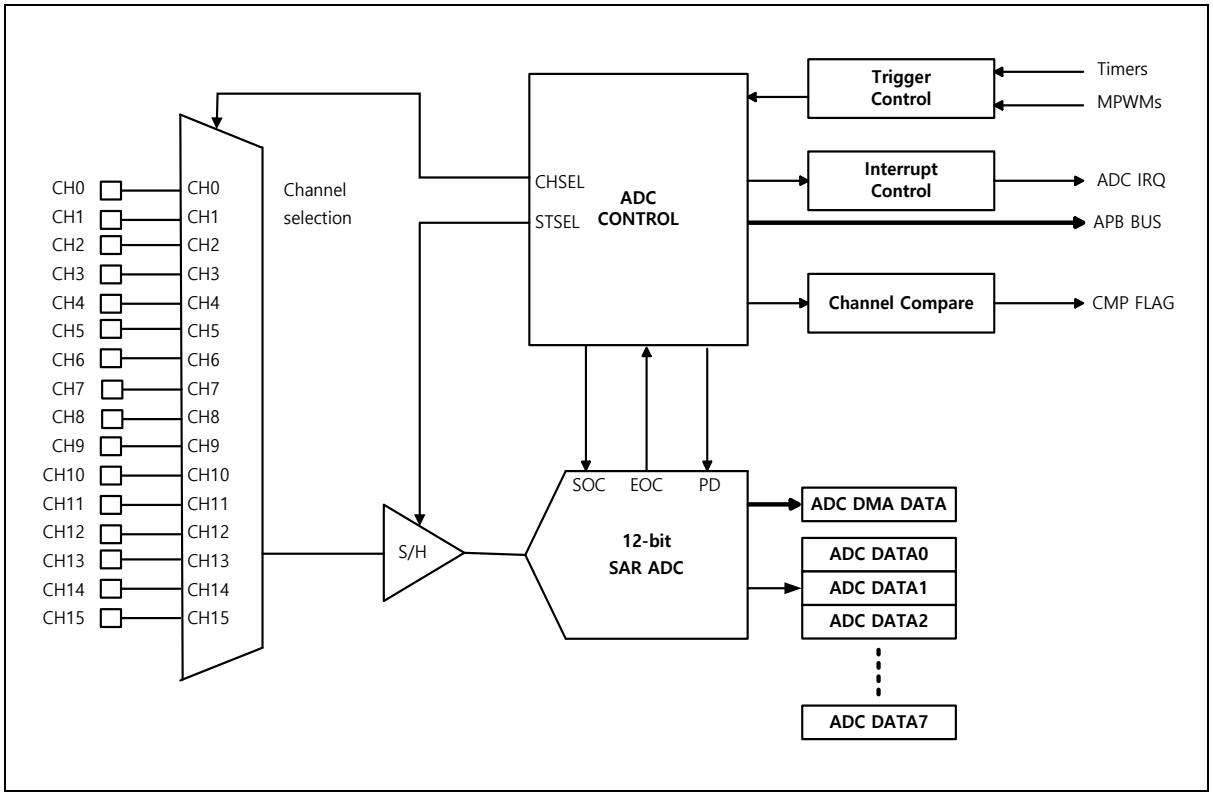


Figure 138. 12-bit ADC Block Diagram

18.2 Registers

Base address of ADC unit is introduced in the followings:

Table 76. Base Address of 12-bit ADC

Name	Base address
ADC	0x4000_3000

Table 77. 12-bit ADC Register Map

Name	Offset	Type	Description	Reset value	Reference
ADC_MR	0x0000	RW	ADC Mode Register	0x0000_0000	18.2.1
ADC_CSCR	0x0004	RW	Current Sequence/Channel Register	0x0000_0000	18.2.2
ADC_CCR	0x0008	RW	ADC Clock Control Register	0x0000_0080	18.2.3
ADC_TRG	0x000C	RW	ADC Trigger Selection Register	0x0000_0000	18.2.4

Table 77. 12-bit ADC Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
ADC_SCSR 1	0x0018	RW	ADC Channel Selection 1 Register	0x0000_0000	18.2.5
ADC_SCSR 2	0x001C	RW	ADC Channel Selection 2 Register	0x0000_0000	18.2.6
ADC_CR	0x0020	RW	ADC Control Register	0x0000_0000	18.2.7
ADC_SR	0x0024	RC	ADC State Register	0x0000_0000	18.2.8
ADC_IER	0x0028	RW	ADC Interrupt Enable Register	0x0000_0000	18.2.9
ADC_DDR	0x002C	RO	ADC DMA Data Register	0x0000_0000	18.2.10
ADC_DR0	0x0030	RO	ADC Sequence 0 Data Register	0x0000_0000	18.2.11
ADC_DR1	0x0034	RO	ADC Sequence 1 Data Register	0x0000_0000	
ADC_DR2	0x0038	RO	ADC Sequence 2 Data Register	0x0000_0000	
ADC_DR3	0x003C	RO	ADC Sequence 3 Data Register	0x0000_0000	
ADC_DR4	0x0040	RO	ADC Sequence 4 Data Register	0x0000_0000	
ADC_DR5	0x0044	RO	ADC Sequence 5 Data Register	0x0000_0000	
ADC_DR6	0x0048	RO	ADC Sequence 6 Data Register	0x0000_0000	
ADC_DR7	0x004C	RO	ADC Sequence 7 Data Register	0x0000_0000	
ADC_CMPR	0x0070	RW	ADC Channel Comparator Register	0x0000_0000	18.2.12

18.2.1 ADC_MR: ADC mode register

ADC_MR is a mode setting register for the ADC Module. This register must be set first for the intended use of the ADC Module.

ADC_MR=0x4000_3000

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		TRGINFO	CHINFO	Reserved	DMAEN		STSEL	Reserved	SEQCNT	ADEN	ARST	ADMOD	Reserved	TRGSEL
-		0	0	-	0		0x00	-	000	0	0	00	-	00
-		RW	RW	-	RW		RW	-	RW	RW	RW	RW	-	RW

21	TRGINFO	Trigger information option (In external trigger mode)		
		0	Option disable	
		1	Trigger source information will be stored in ADCDR[31:24]	
20	CHINFO	Channel information option		
		0	Option disable	
		1	Converted channel information will be stored in ADCDR[20:16]	
17	DMAEN	DMA Enable bit (Must be set When ADEN =1)		
		When the DMA function is enabled and the ADC receives a completion signal from the DMAC, a DMA request is generated at the interrupt request is generated or conversion is finished. (Include Burst Mode)		
16 12	STSEL	Sampling Time Selection		
		It is a value to set the time window section that can recognize the Next Trigger. The applied point is applied immediately after Trigger. ADC Sampling Time is (2+STSEL [4:0]) MCLK cycle. Minimum Sampling Time is 2 MCLK. When STSEL[4:0] = b'11111, Sampling Channel always enable.		
10 8	SEQCNT	Number of conversion in a sequence If ADMOD[5:4] is 0 and SEQCNT[10:8] isn't 0, CSEQN increase to SEQCNT by trigger event. (SEQCNT apply only in Single/Sequential Mode)		
		000	Single Mode	100 5 Sequence ADC
		001	2 Sequence ADC	101 6 Sequence ADC
		010	3 Sequence ADC	110 7 Sequence ADC
		011	4 Sequence ADC	111 8 Sequence ADC
7	ADEN	ADC Enable bit		
		0	Disable	
		1	Enable	
6	ARST	After sequence finish, restart bit.		
		0	Stop after finish (ASTART must be set to 1 to restart)	
		1	Restart after finish	

5 4	ADMOD	ADC Mode Selection bit.	
		00	Single/Sequential Conversion Mode
		01	Burst Conversion Mode
		10	Multiple Conversion Mode
		11	No effect
1 0	TRGSEL	Trigger Selection bit.	
		00	Event Trigger Disable/Soft Trigger Only
		01	Timer Event Trigger/Soft Trigger
		10	Reserved
		11	Reserved

NOTES:

1. When ADCMOD is set to Burst Conversion Mode, the ADC channel is controlled from BST0CH to BST7CH. Burst Mode always starts at BST0CH. (In 3-busrt mode, the analog input of the channel assigned in BST0CH / BST1CH / BST2CH is sequentially converted.)
2. If it is set to Multiple Mode, it will be converted as soon as it triggers regardless of the order of Trigger set in TRG register.

18.2.2 ADC_CSCR: ADC current sequence/channel register

ADC_CSCR consists of the Current Sequence Number and Current Active Channel value. CSEQN (Current Sequence Number) can set the Current Sequence Number immediately. This is a 16-bit register. In addition, the ADEN bit of the ADC_MR Register must be set before this register is set.

ADC_CSCR=0x4000_3004

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			CSEQN	Reserved	CACH
-			000	-	0x00
-			RW	-	RO

10 8	CSEQN	Current Sequence Number,	
		000	Current Sequence is 0
		001	Current Sequence is 1
		010	Current Sequence is 2
		011	Current Sequence is 3
		100	Current Sequence is 4
		101	Current Sequence is 5
		110	Current Sequence is 6
		111	Current Sequence is 7

4 0	CACH	Current Active Channel	
		00000	ADC channel 0 is active
		00001	ADC channel 1 is active
		00010	ADC channel 2 is active
		00011	ADC channel 3 is active
		00100	ADC channel 4 is active
		00101	ADC channel 5 is active
		00110	ADC channel 6 is active
		00111	ADC channel 7 is active
		01000	ADC channel 8 is active
		01001	ADC channel 9 is active
		01010	ADC channel 10 is active
		01011	ADC channel 11 is active
		01100	ADC channel 12 is active
		01101	ADC channel 13 is active
		01110	ADC channel 14 is active
		01111	ADC channel 15 is active
		Others	Reserved

18.2.3 ADC_CCR: ADC clock control register

ADC_CCR is a clock control register of ADC Module. This is a 16-bit register.

ADC_CCR=0x4000_3008

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		ADCPDA	CLKDIV	ADCPD	Reserved	CLKINVT	Reserved
-		0	0x00	1	-	0	-
-		RW	RW	RW	-	RW	-

15	ADCPDA	ADC R DAC disable to save power	
		Don't set '1' here (it's optional bit)	
14 8	CLKDIV	ADC clock division value bit(When EXTCLK is 0, CLKDIV Enable)	
		- CLKDIV=0 → ADC Clock = ADC Input Clock (Bypass) - CLKDIV=1 → ADC Clock = Clock Stop - CLKDIV≥2 → ADC Clock = ADC Input Clock / CLKDIV **NOTE:** In continuous conversion mode or burst conversion mode, the CLKDIV must be set to 3 or higher When ADC clock is divided and CLKDIV value is set to 2 or more, ADC clock should be set not to exceed 24MHz.	
7	ADCPD	ADC Deep sleep	
		0	ADC Normal Mode
		1	ADC Deep sleep Mode
5	CLKINVT	Divide Clock Inversion (Option bit)	
		0	Duty ratio of divided clock is larger than 50%
		1	Duty ratio of divided clock is less than 50%

18.2.4 ADC_TRG: ADC trigger selection register

This Register is ADC Trigger Selection Register of ADC Module.

ADC_TRG=0x4000_300C

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
SEQTRG7	SEQTRG6	SEQTRG5	SEQTRG4	SEQTRG3	SEQTRG2	SEQTRG1	SEQTRG0 BSTTRG
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

30 28	SEQTRG7	8th Sequence Trigger Source
27 24	SEQTRG6	7th Sequence Trigger Source
23 20	SEQTRG5	6th Sequence Trigger Source
19 16	SEQTRG4	5th Sequence Trigger Source
15 12	SEQTRG3	4th Sequence Trigger Source
11 8	SEQTRG2	3th Sequence Trigger Source
7 4	SEQTRG1	2ed Sequence Trigger Source
3 0	SEQTRG0 BSTTRG	1st Sequence Trigger Source Burst Conversion Trigger Source

NOTE: 1st sequence is the highest priority in multi-mode, and 8th sequence is the lowest priority in multi-mode.

Table 78. Trigger Source of ADC

Value	Timer (TRGSEL[1:0] = 0x1)
0	TIMER 10
1	TIMER 11
2	TIMER 12
3	TIMER 13
4	TIMER 20
5	TIMER 21
6	TIMER 30
7	TIMER 40
8	
9	
10	
11	
15	ASTART

NOTES:

1. ASTART is a S/W trigger in the ADC_CR register.
2. In order to use Timer30 as a trigger, TIMER30_ADTCR and TIMER30_ADTDR must be set.

18.2.5 ADC_SCSR1: ADC channel selection 1 register

ADC_SCSR1 is a 32-bit register. Each selected channel is associated with a trigger selection register. When ADEN of ADC_MR Register set, Can write in ADC_SCSR1.

ADC_SCSR1=0x4000_3018

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	SEQ3CH	Reserved	SEQ2CH	Reserved	SEQ1CH	Reserved	SEQ0CH
-	0x0	-	0x0	-	0x0	-	0x0
-	RW	-	RW	-	RW	-	RW

28 24	SEQ3CH	4th Conversion Sequence Channel Selection
20 16	SEQ2CH	3th Conversion Sequence Channel Selection
12 8	SEQ1CH	2th Conversion Sequence Channel Selection
4 0	SEQ0CH	1th Conversion Sequence Channel Selection

18.2.6 ADC_SCSR2: ADC channel selection 2 register

ADC_SCSR2 is a 32-bit register. Each selected channel is associated with a trigger selection register. When ADEN of ADC_MR Register set, Can write in ADC_SCSR2.

ADC_SCSR2=0x4000_301C

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	SEQ7CH	Reserved	SEQ6CH	Reserved	SEQ5CH	Reserved	SEQ4CH
-	0x0	-	0x0	-	0x0	-	0x0
-	RW	-	RW	-	RW	-	RW

28 24	SEQ7CH	8th Conversion Sequence Channel Selection
20 16	SEQ6CH	7th Conversion Sequence Channel Selection
12 8	SEQ5CH	6th Conversion Sequence Channel Selection
4 0	SEQ4CH	5th Conversion Sequence Channel Selection

18.2.7 ADC_CR: ADC control register

ADC_CR is a register for ADC Control.

ADC_CR=0x4000_3020

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			ASTOP	Reserved	TRGCLR	ASTART
-			0	-	0	0
-			WO	-	RW	RW

7	ASTOP	ADC STOP bit	
		0	No Effect
		1	ADC conversion stop (will be clear next @ADC clock) If ASTOP set after conversion cycle start, present conversion would be completed.
1	TRGCLR	ADC all trigger flags cleared option	
		0	No clear
		1	Clear all trigger flags of previous ADC operation
0	ASTART	ADC START bit	
		0	No ADC Conversion
		1	ADC Conversion Start (will be clear next @ADC clock) ADEN should be "1" to start ADC If ASTART is set as '1' when ARST is '0' in trigger event mode, ADC conversion will start once as SEQCNT set.

18.2.8 ADC_SR: ADC state register

ADC_SR is a register for indicating the ADC state.

ADC_SR=0x4000_3024

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			COMPIFLG	Reserved		DOVRUN	DMAF	TRGIF	EOSIF	Reserved	EOCIF
-			0	-		0	0	0	0	-	0
-			RC	-		RO	RO	RC	RC	-	RC

8	COMPIFLG	Compare Interrupt Flag bit	
		0	No Interrupt occurred
		1	Interrupt occurred (Write '1' to Clear Flag)
5	DOVRUN	DMA Overrun Flag (Not Interrupt)	
		0	No Flag occurred
		1	Flag occurred
4	DMAF	DMA Done Received Flag (DMA transfer is completed)	
		0	No Flag occurred
		1	Flag occurred
3	TRGIF	ADC Trigger Interrupt Flag	
		0	No Flag occurred
		1	Flag occurred (Write '1' to Clear Flag)
2	EOSIF	Sequence End Interrupt Flag	
		0	No Flag occurred
		1	Flag occurred (Write '1' to Clear Flag)
0	EOCIF	Sequence Conversion End Interrupt Flag	
		0	No Flag occurred
		1	Flag occurred (Write '1' to Clear Flag)

NOTE:
Flag check of polling method uses EOCIF bit.

Example code)

```
while ((ADC_SR & 0x01) == 1)        // EOCIF Flag Checking
{
        ADCSR = 0x1;                // EOCIF Flag Clear
}
```

18.2.9 ADC_IER: ADC interrupt enable register

ADC_IER is a register to enable ADC Interrupt.

ADC_IER=0x4000_3028

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				DMAIE	TRGIE	EOSIE	Reserved	EOCIE
-				0	0	0	-	0
-				RW	RW	RW	-	RW

4	DMAIE	DMA Done Interrupt Enable	
		0	Disable
		1	Enable
3	TRGIE	ADC Trigger Conversion Interrupt Enable	
		0	Disable
		1	Enable
2	EOSIE	ADC Sequence Conversion Interrupt Enable	
		0	Disable
		1	Enable
0	EOCIE	ADC Single Conversion Interrupt Enable	
		0	Disable
		1	Enable

18.2.10 ADC_DDR: ADC DMA data register

ADC_DDR is a register of ADC DMA Conversion result.

ADC_DDR=0x4000_302C

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
TRGINFO7	TRGINFO6	TRGINFO5	TRGINFO4	TRGINFO3	TRGINFO2	TRGINFO1	TRGINFO0	Reserved	ADMACH	ADC DMA Temporary Data	Reserved
0	0	0	0	0	0	0	0	-	0x00	0x0000	-
RO	RO	RO	RO	RO	RO	RO	RO	-	RO	RO	-

31 24	TRGINFOx (x=0~7)	ADC Trigger Information (ADC Trigger Information is Trigger source captured at EOC time.) *Must be enable TRGINFO function of ADC_MR Register. Multiple Mode) The lower the x of TRGINFOx, the higher the priority. Multiple TRGINFO bits can be read as 1 if Pending by another trigger. Single/Sequential Mode) It can reference the source Pending by another trigger. The currently processed trigger source must refer to CSEQN of CSCR register.

20 16	ADMACH	DMA ADC Channel Indicator *Enable the CHINFO function of ADC_MR register.
15 4	ADDMAR	DMA ADC Conversion Result Data (12-bit)

NOTE: Even when the DMA function is not used, data is temporarily stored in this buffer before the data is stored in the corresponding buffer, and channel information of the data can be also known.

18.2.11 ADC_DR: ADC sequence 0 to 7 data register

ADC_DR indicates the result of the ADC Conversion. There are eight of these registers.

ADC_DR0=0x4000_3030, ADC_DR1=0x4000_3034, ADC_DR2=0x4000_3038, ADC_DR3=0x4000_303C
ADC_DR4=0x4000_3040, ADC_DR5=0x4000_3044, ADC_DR6=0x4000_3048, ADC_DR7=0x4000_304C

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
TRGINFO7	TRGINFO6	TRGINFO5	TRGINFO4	TRGINFO3	TRGINFO2	TRGINFO1	TRGINFO0	Reserved	ACH	ADDATA	Reserved
0	0	0	0	0	0	0	0	-	0x00	0x000	-
RO	RO	RO	RO	RO	RO	RO	RO		RO	RO	-

31 24	TRGINFOx (x=0~7)	ADC Trigger Information (ADC Trigger Information is Trigger source captured at EOC time.) *Must be enable TRGINFO function of ADC_MR Register. Multiple Mode) The lower the x of TRGINFOx, the higher the priority. Multiple TRGINFO bits can be read as 1 if Pending by another trigger. Single/Sequential Mode) It can reference the source Pending by another trigger. The currently processed trigger source must refer to CSEQN of CSCR register.
20 16	ACH	ADC Channel Information * Enable the CHINFO function of ADC_MR register.
15 4	ADDATA	ADC Input data

18.2.12 ADC_CMPR: ADC channel compare register

ADC_CMPR is an ADC channel compare control register.

ADC_CMPR=0x4000_3070

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	COMPIEN	COMPEN	Reserved	LTE	CCH	CVAL	Reserved
-	0	0	-	0	00000	0x000	-
-	RW	RW	-	RW	RW	RW	-

24	COMPIEN	Compare Interrupt Enable bit	
		0	Disable
		1	Enable
23	COMPEN	Compare Operation Enable bit	
		0	Disable
		1	Enable
21	LTE	AD Conversion Value Output Timing Setting	
		0	When ADC > CVAL, Output
		1	When ADC ≤ CVAL, Output
20 16	CCH	Compare Channel	
		00000	Compare Channel is ADC channel 0
		00001	Compare Channel is ADC channel 1
		00010	Compare Channel is ADC channel 2
		00011	Compare Channel is ADC channel 3
		00100	Compare Channel is ADC channel 4
		00101	Compare Channel is ADC channel 5
		00110	Compare Channel is ADC channel 6
		00111	Compare Channel is ADC channel 7
		01000	Compare Channel is ADC channel 8
		01001	Compare Channel is ADC channel 9
		01010	Compare Channel is ADC channel 10
		01011	Compare Channel is ADC channel 11
		01100	Compare Channel is ADC channel 12
		01101	Compare Channel is ADC channel 13
		01110	Compare Channel is ADC channel 14
		01111	Compare Channel is ADC channel 15
		10000	Compare Channel is ADC channel 16
		Others	Reserved
15 4	CVAL	Compare Value Bit	

18.3 Functional description

18.3.1 ADC single mode timing diagram

When ADC_MR.ADMOD is 0x0 and ADC_MR.SEQCNT is 0x0, The ADC conversion starts when the CR.ASTART bit is set to '1'. When ADC_CR.ASTART is set, SOC (start of conversion) is active at 3 ADC clocks and ADC_SR.EOCIRQ is set at 2 ADC clocks and 2 PCLKs after the end of conversion.

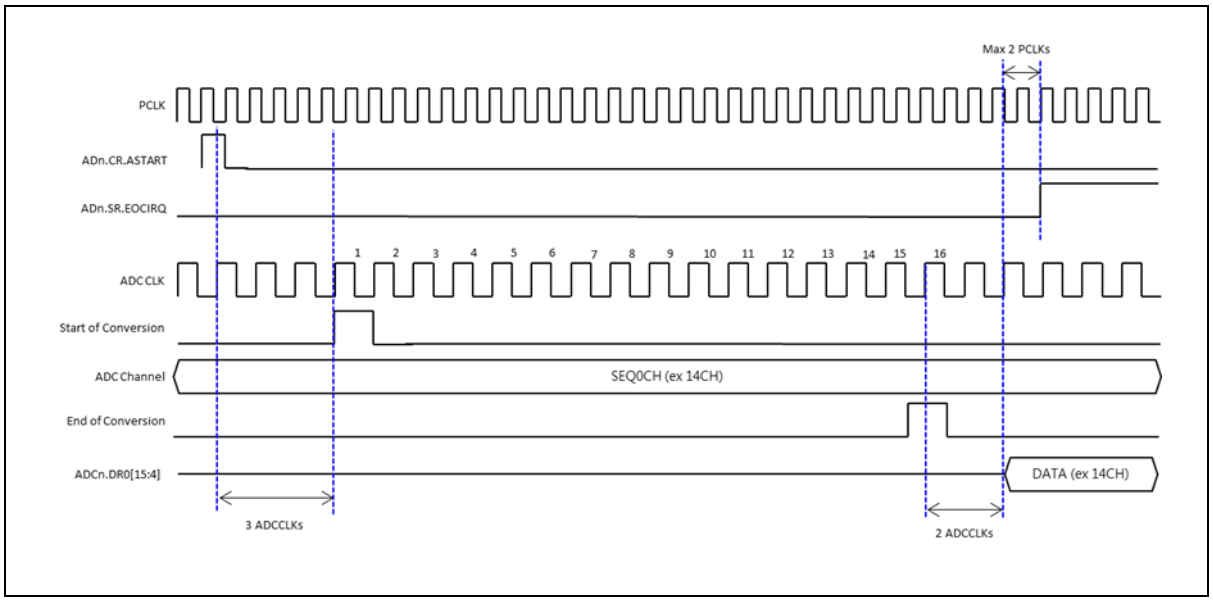


Figure 139. ADC Single Mode Timing (When ADC.MR.AMOD = '0')

18.3.2 ADC burst mode timing diagram

There are two source for creating SOC in Burst Mode. First, TRG event (timer and MPWM) and ASTART. If TRGSEL is set to Timer Event Trigger or MPWM Event Trigger, the SOC will be the trigger of TRG.BSTTRG. For example, if TRG.BSTTRG is set to TIMER3, ADC Conversion is started by the trigger of TIMER3. When a trigger event of BSTTRG occurs, the ADC will change the ADC channel to the value set by MR.SEQCNT. See Figure 140.

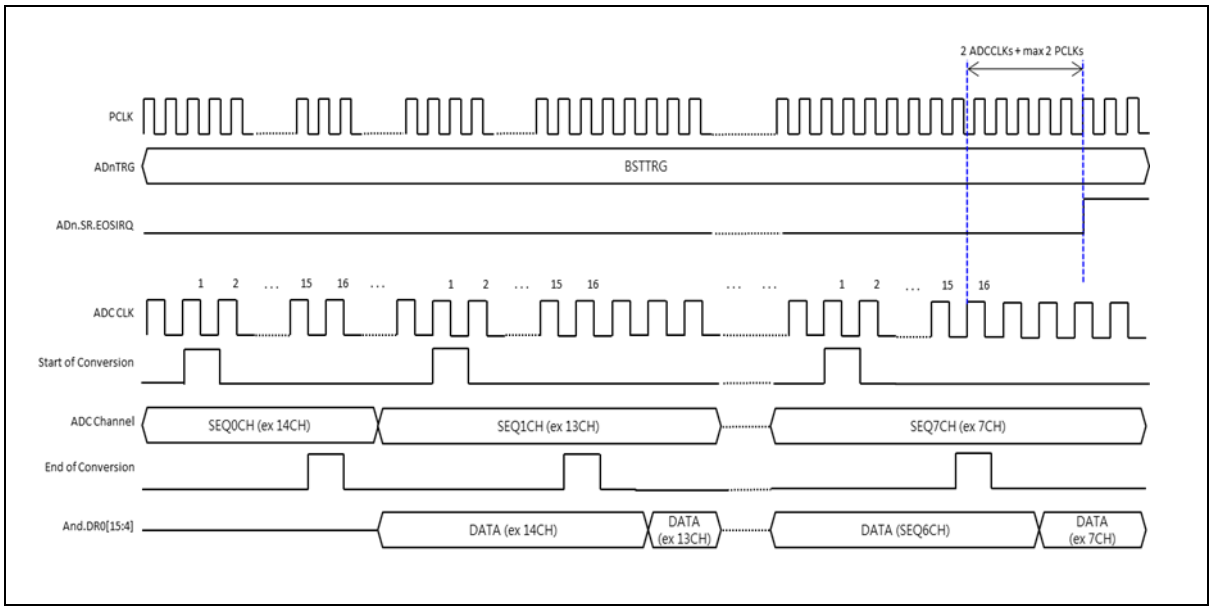


Figure 140. ADC Burst Mode Timing (When ADC.MR.AMOD = '1')

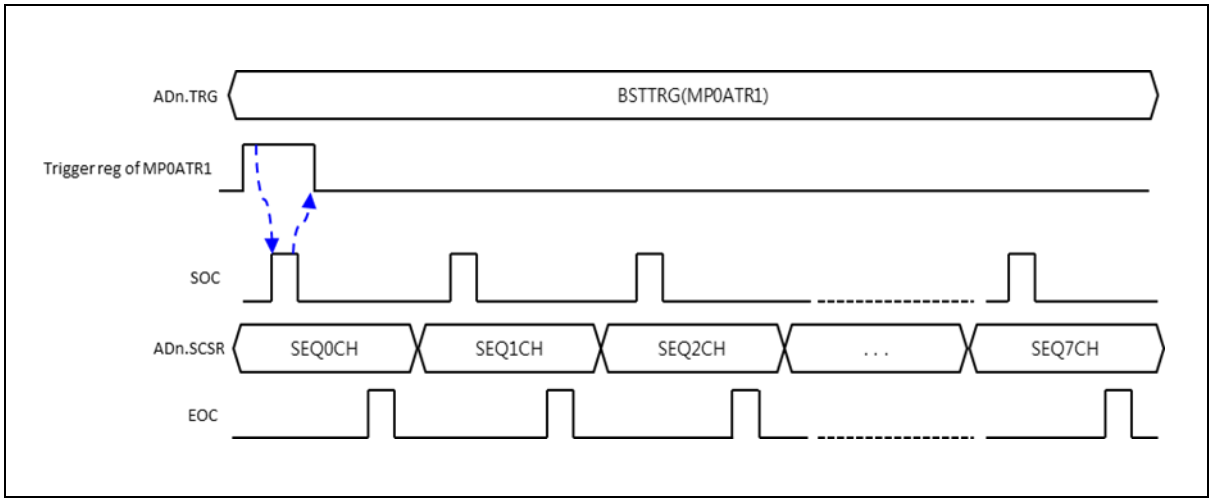


Figure 141. ADC Trigger Timing in Burst Mode (SEQCNT = 3'b111, 8 Sequential Conversion)

18.3.3 ADC sequential mode timing diagram

To set sequential mode, ADC_MR.ADMOD is 2'b00 and ADC_MR.SEQCNT must not be 3'b000. The operation of the sequential mode is the almost same as burst mode. Difference is the source of the SOC. Each SOC is created by a trigger of SEQTRGx with each SEQCNT. See Figure 142.

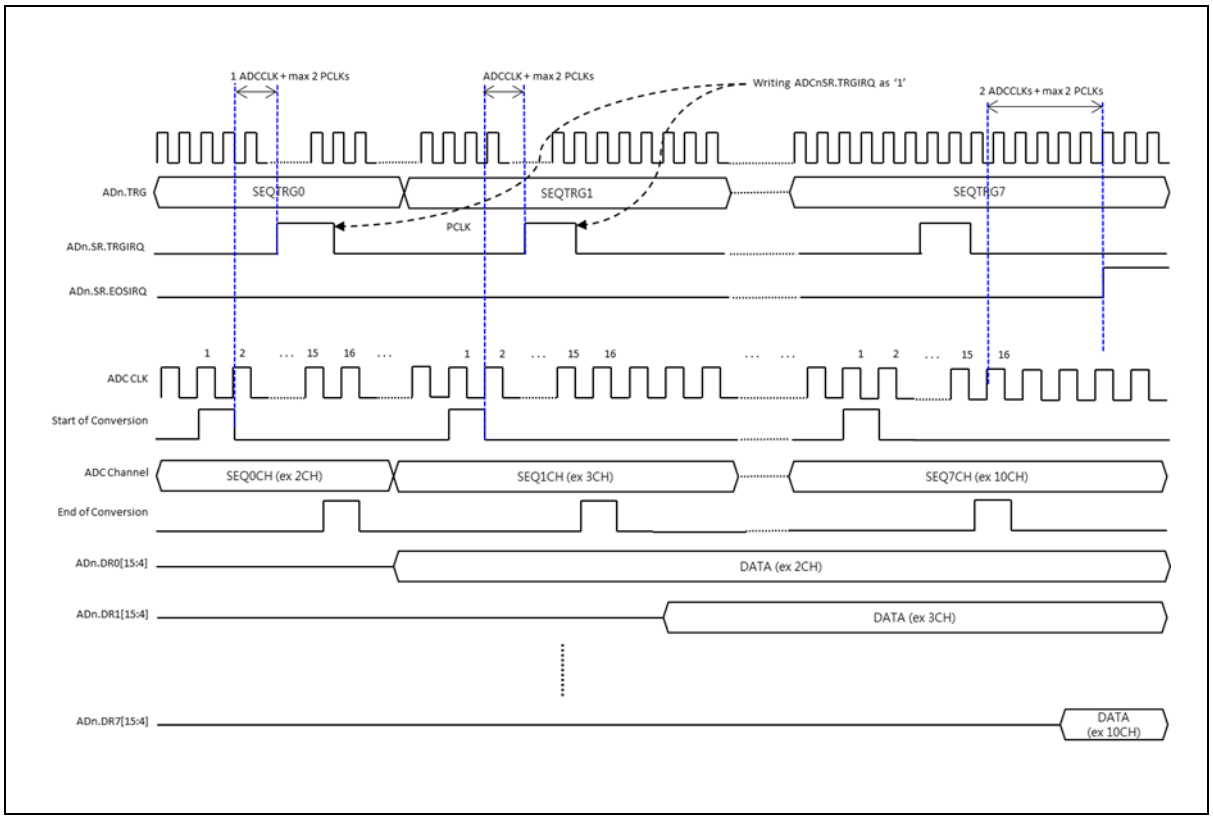


Figure 142. ADC Sequential Mode Timing (When MR.AMOD = '0' and MR.SEQCNT ≠ '0')

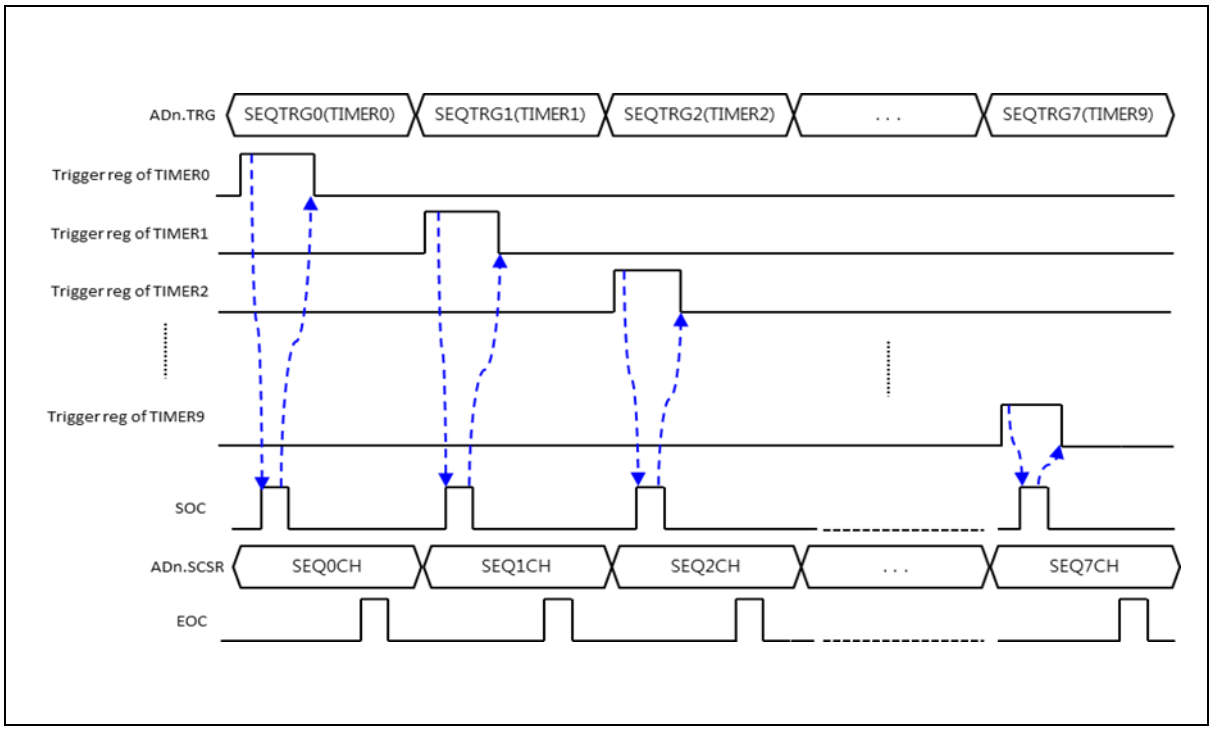


Figure 143. ADC Trigger Timing in Sequential Mode
(SEQCNT = 3'b111, 8 Sequential Conversion)

18.3.4 ADC multiple mode timing diagram

In multiple mode operation, if the desired trigger source is set to TRG.SEQTRG, conversion begins when the corresponding trigger occurs regardless of the order of SEQTRG. For example, if MR.SEQCNT is 3'b011, if one of the four trigger sources (TIMER, MPWM) is selected and CR.ASTART is set, conversion will be processed by the first occurred trigger source, regardless of sequence. It is different from sequential mode and burst mode.

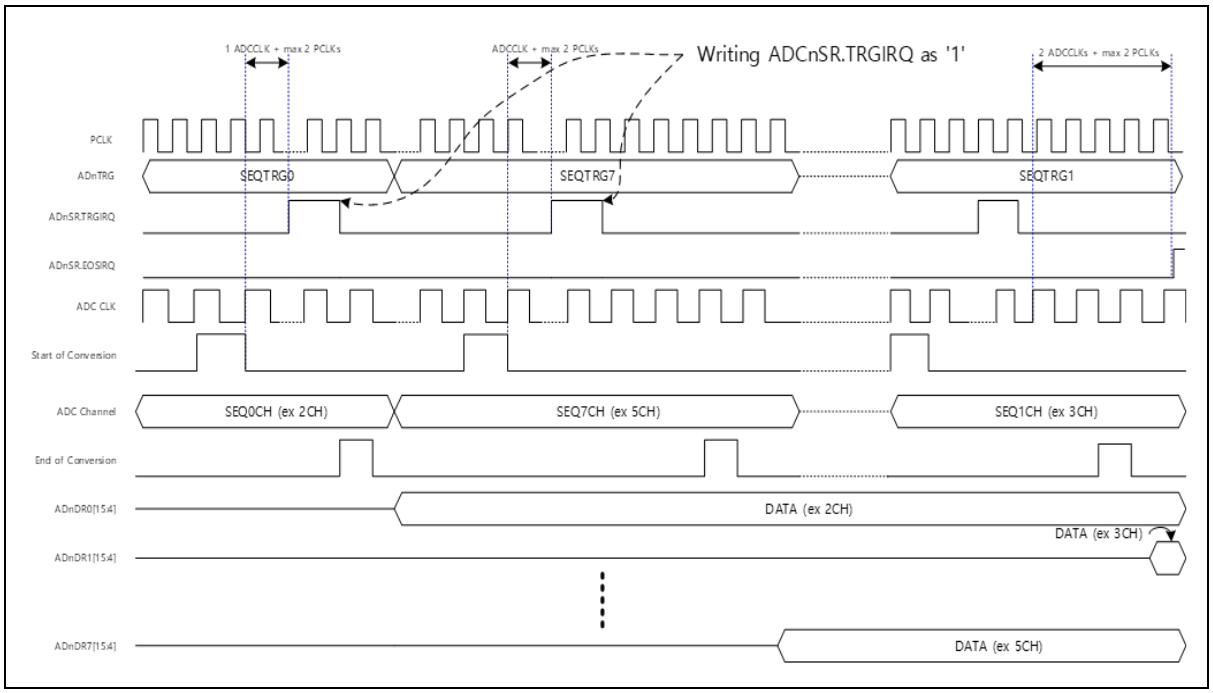


Figure 144. ADC Multiple Mode Timing (When MR.AMOD = '2' and MR.SEQCNT ≠ '0')

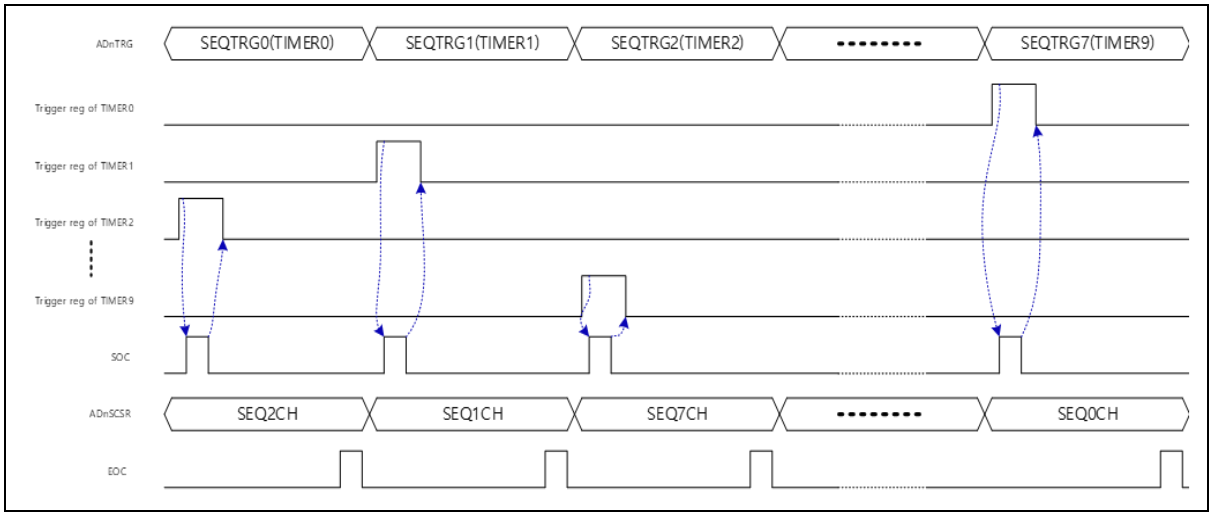


Figure 145. ADC Trigger Timing in Multiple Mode (SEQCNT = 3'b111, 8 Multiple Conversion)

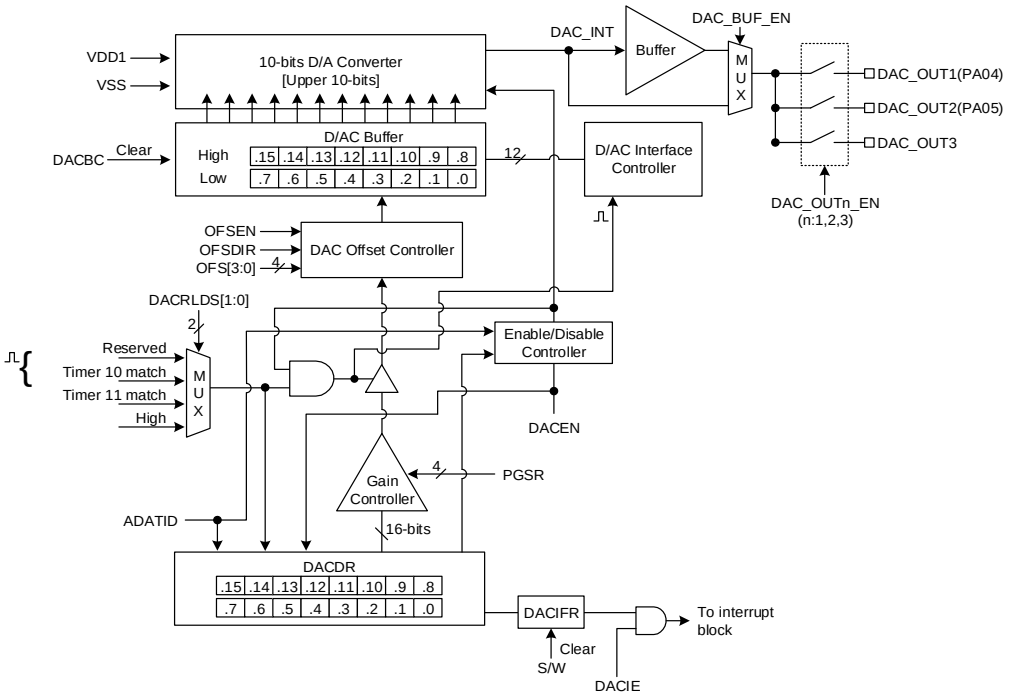
19. 10-bit DAC

Digital-to-analog (D/A) converter uses successive approximation logic to convert 10-bit digital value to an analog output level.

DAC module has six registers which are the DAC control register (DACCR), DAC data high register (DACDRH), DAC data low register (DACDRL), DAC buffer high register (DACBRH), DAC buffer low register (DACBRL) and programmable gain selection register (PGSR).

19.1 10-bit DAC block diagram and analog power pin

In Figure 146, 10-bit DAC is described in a block diagram.



NOTES:

1. Gain controller converts data in the sequence listed below:
   A. Get the DACDR[15:6] value.
   B. Adjust the data by selection gain.
   C. Write the data to the D/AC buffer.
2. DAC interface block doesn't work when DACRLDS bits of the DACCR register are set to "00b".

Figure 146. 10-bit DAC Block Diagram

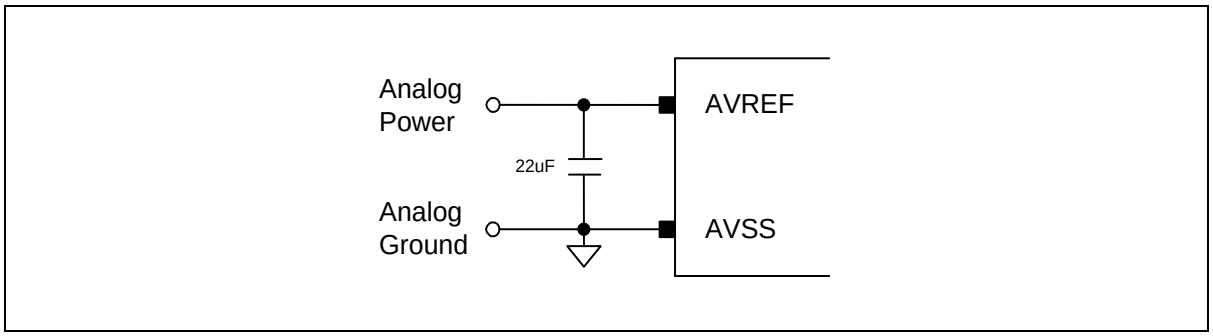


Figure 147. Analog Power (AVREF) Pin with a Capacitor

19.2 Registers

Base address of DAC unit is introduced in the followings:

Table 79. Base Address of DAC

Name	Base address
DAC	0x4000_3500

Table 80. 10-bit DAC Register Map

Name	Offset	Type	Description	Reset value	Reference
DACDR	0x0000	RW	DAC data high register	0x0000_0000	19.2.1
DACBR	0x0004	RO	DAC buffer high register	0x0000_0000	19.2.2
DACCR	0x0008	RW	DAC control register	0x0000_0000	19.2.3
PGSR	0x000C	RW	Programmable gain selection register	0x0000_0005	19.2.4
DAOFSCR	0x0010	RW	DAC offset control register	0x0000_0000	19.2.5

19.2.1 DAC_DR: DAC data register

DAC_DR=0x4000_3500

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		DACDR	Reserved
-		000	-
-		RW	-

13 4	DACDR	DAC Data (10-bit) The DACDR[13:4] is a binary format.

19.2.2 DAC_BR: DAC buffer register

DAC_BR=0x4000_3504

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		DACBR	Reserved
-		000	-
-		R	-

13 4	DACBR	DAC Buffer Data (10-bit) The DACBR[13:4] is a binary format.

19.2.3 DAC_CR: DAC control register

DAC_CR=0x4000_3508

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			DAC_OUT3_EN	DAC_OUT2_EN	DAC_OUT1_EN	DAC_BUF_EN	Reserved	REFSEL	DACIE	DACIFR	ADATID	DACBC	Reserved	DACRLDS		DACEN
-			0	0	0	0	-	0	0	0	0	0	-	0		0
-			RW	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW		RW

13	DAC_OUT3_EN	DAC_OUT3 output enable **NOTE:** DAC_OUT3 isn't connected to Pin.	
		0	Disable
		1	Enable
12	DAC_OUT2_EN	DAC_OUT2 output enable **NOTE:** DAC_OUT2 is connected to PA5. Therefore, when using DAC_OUT2, PA5 cannot be used as GPIO	
		0	Disable
		1	Enable
11	DAC_OUT1_EN	DAC_OUT1 output enable **NOTE:** DAC_OUT1 is connected to PA4. Therefore, when using DAC_OUT1, PA4 cannot be used as GPIO	
		0	Disable
		1	Enable
10	DAC_BUF_EN	DAC Buffer Selection	
		0	Disable (buffer bypass)
		1	Enable

8	REFSEL	DAC Reference Selection	
		0	AVDD (Internal)
		1	DAVREF Pin (External)
7	DACIE	Enable or Disable D/AC Interrupt	
		0	Disable
		1	Enable
6	DACIFR	When D/AC Interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. This interrupt is for a result that the DACDRH register automatically increments to "800xH" or decrements to "000xH". Write '1' has no effect.	
		0	D/AC interrupt no generation
		1	D/AC interrupt generation
5	ADATID	Automatically DAC Data Increment/Decrement	
		0	Disable automatically D/AC data increment/decrement
		1	Automatically D/AC data increment from DACDR value to "800xH" when DACEN bit is changed to "1b". Automatically D/AC data decrement from DACDR value to "000xH" when DACEN bit is changed to "0b". NOTE) It doesn't fetch data from FADPCM block during automatically data increment/decrement even if the FADFEN bit is '1'
4	DACBC	DAC Buffer Clear	
		0	No effect.
		1	Clear the D/AC buffer (When write, automatically cleared to '0' after being cleared)
2 1	DACRLDS	DAC Reload Selection. These bits select a reload signal to load data from D/AC data register to buffer.	
		00	Always
		01	Reserved
		10	Timer 10 match signal
		11	Timer 11 match signal
0	DACEN	DAC Enable Bit	
		0	Stop D/AC operation (Low level output)
		1	Start D/AC operation

19.2.4 DAC_PGSR: Programmable gain control register

DAC_PGSR=0x4000_350C

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				PGS3	PGS2	PGS1	PGS0
-				0	1	0	1
-				RW	RW	RW	RW

3 0	PGS	Programmable Gain Selection	
		0000	-30dB
		0001	-24dB
		0010	-18dB
		0011	-12dB
		0100	-6dB
		0101	0dB
		0110	+6dB
		0111	+12dB
		1000	+18dB
		1001	+24dB
		1010	+30dB
		Others	Reserved

19.2.5 DAC_OFSCR: DAC offset control register

DAC_OFSCR=0x4000_3510

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			OFSEN	OFSDIR	Reserved	OFS
-			0	0	-	0
-			RW	RW	-	RW

7	OFSEN	D/AC Offset Control Enable Bit	
		0	Disable
		1	Enable
6	OFSDIR	D/AC Offset Direction Selection Bit	
		0	D/AC buffer data are subtracted by (n + 1)
		1	D/AC buffer data are added by (n + 1)
		NOTE: Where n is the OFS value and n = 0, 1, 2, ,,,, 15.	
3 0	OFS	D/AC Offset Value	

19.3 Functional description

DAC is R-2R in structure and its data register is a binary in format. It is possible to update the DAC data register as the result output data (DODRH/L register) of FADPCM decoder every the FADPCM decoder result signal by FADFEN bit set to "1b". The 16-bit digital value of the D/AC data register goes into D/AC buffer register through the programmable gain controller every a reload signal. The reload signal is one of the "Always", "FADPCM decoder match signal", "Timer 0 match signal" and "Timer 1 match signal". The signal is selected by the DACRLDS[1:0] bits.

The programmable gain controller has eleven step (-30dB, -24dB, -18dB, -12dB, -6dB, 0dB, +6dB, +12dB, +18dB, +24dB and +30dB) and the gain is selected by the programmable gain register (PGSR).

The value of the DAC data register can be automatically incremented from the current data value to "800xH" when the DAC is enabled by DACEN bit set to "1b" and vice versa. At that time, the DAC interrupt flag bit (DACIFR) is set to "1b". The auto-increment/decrement structure for DAC data is useful to remove a pop noise when a speaker is turn on/off. Two kinds External D/AC Converter is be accessed

19.3.1 DAC data buffer register

DAC data and buffer registers are 16-bits, respectively. But only the upper 12-bits of the DAC buffer register specifies to generate DAC output signal. The reset value of the data and buffer is "0000H". The DAC output value, VDAC, is calculated by the following formula:

VDAC = VDD X (n ÷ 4096), (n = 0, 1, 2,,,,, 4095. That is DACBR[15:4] value)

19.3.2 Automatic DAC data increment/decrement

"Automatic DAC data increment/decrement" function is important when removing a pop noise. If this function is not embedded, a programmer has to code to reduce a pop noise on a speaker. In A31G32x series, DACDR[15:6] value increases a current DAC data value to 800H with "automatic D/AC data increment" when ADATID bit is set to "1b" and DACEN bit is changed to "1b" from "0b". DACDR[15:6] value decreases a current D/AC data value to 000H with "automatic D/AC data decrement" when ADATID bit is set to "1b" and DACEN bit is changed to "0b" from "1b".

Steps below introduce a procedure removing a pop noise when a speaker is turn on:

1. Write "05H" for 0dB to PGSR register.
2. Write "0000H" to DAC data register.

4. Clear DAC buffer register by DACBC bit set to "1b".
5. Select one of the "Timer 0 match signal" and "Timer 1 match signal" for DAC reload signal with DACRLDS[1:0] bits.
6. Start D/AC operation by DACEN bit set to "1b".

Steps below introduce a procedure removing a pop noise when a speaker is turned off:

1. Write "05H" for 0dB to PGSR register.
2. Keep the value of DACRLDS[1:0] bits set when speaker is turn on.
3. Stop D/AC operation by DACEN bit set to "0b".

19.3.3 Programmable gain controller

There are 11 selectable levels of gain in the programmable gain controller. The levels are -30dB, -24dB, -18dB, -12dB, -6dB, 0dB, +6dB, +12dB, +18dB, +24dB and +30dB. The gain is selected by configuring programmable gain register (PGSR).

The gain controller converts value of DACDR[15:0] by following the procedure introduced below:

1. Get the value of DAC data register (DACDR[15:0]).
2. Change the data into a 16-bit signed format.
3. Adjust the data by a selected gain.
4. Modify the data into a 16-bit binary format.
5. Write the data to the DAC buffer register (DACBR[15:0]).

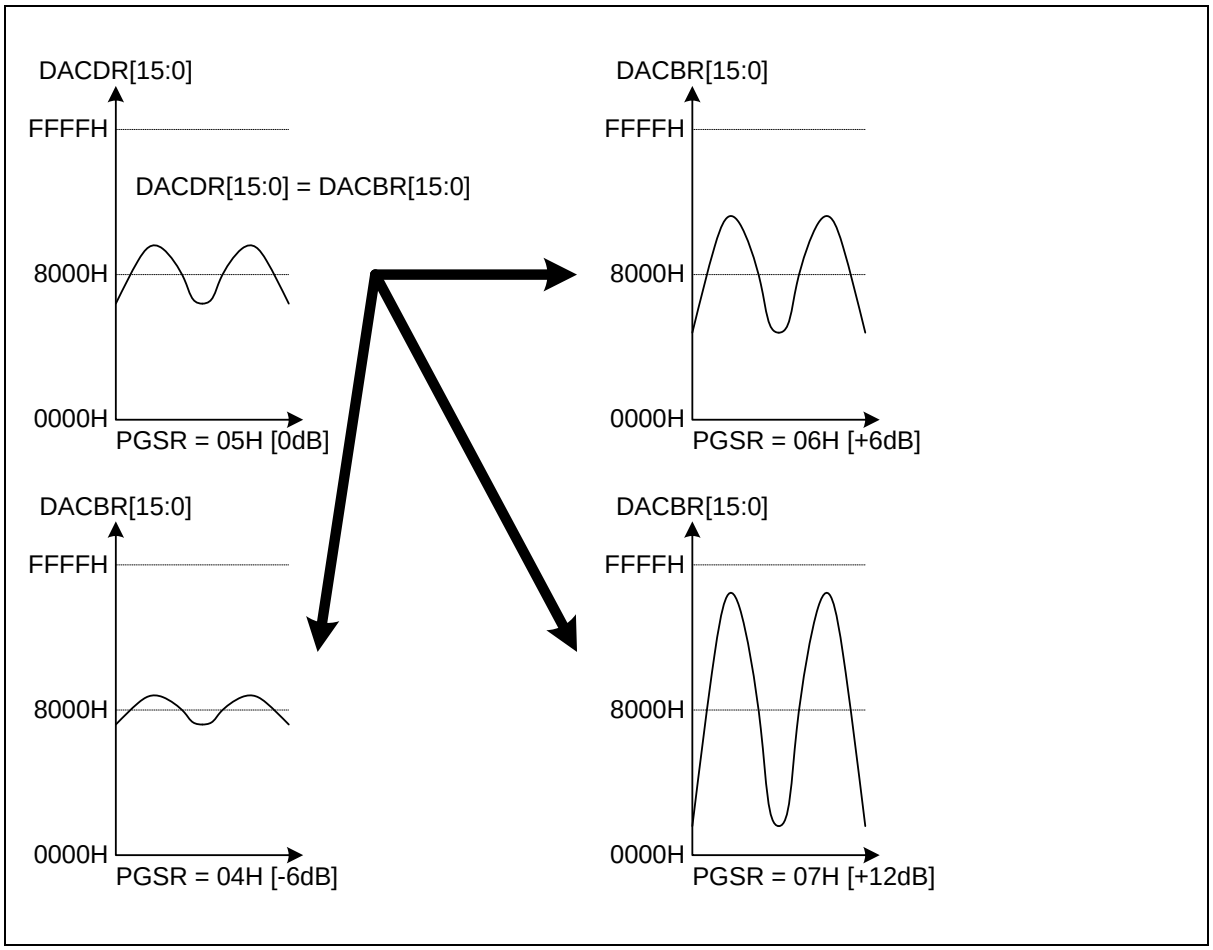


Figure 148. DACBR[15:0] Value by a Selected Gain

20. Comparator

Comparator of A31G32x series compares one analogue voltage level with external reference voltage, or internal reference voltage output voltage.

The comparator features the followings:

- 2 Comparators
- Internal BGR reference for comparator
- Comparator output de-bounce function
- Level and edge interrupt mode support for comparator

20.1 Comparator block diagram

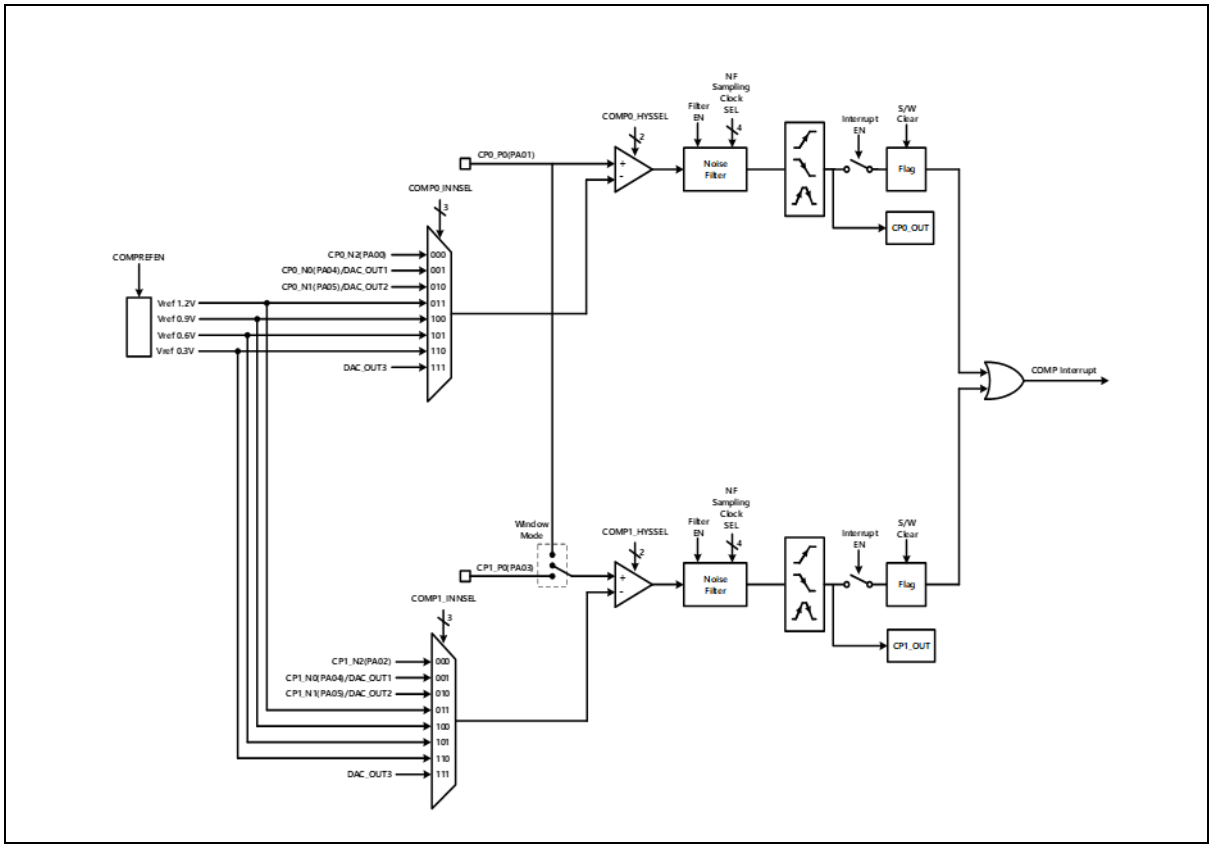


Figure 149. Comparator Block Diagram

20.2 Registers

Base address of the comparator unit is introduced in the followings:

Table 81. Base Address of Comparator

Name	Base address
COMP	0x4000_3420

Table 82. Comparator Register Map

Name	Offset	Type	Description	Reset value	Reference
COMP0CR	0x0000	RW	Comparator 0 Control Register	0x0000_0000	20.2.1
COMP1CR	0x0004	RW	Comparator 1 Control Register	0x0000_0000	20.2.2
COMPDBNC	0x0010	RW	Comparator Debounce Register	0x0000_0000	20.2.3
COMPICON	0x0014	RW	Comparator Interrupt Control Register	0x0000_0000	20.2.4
COMPIEN	0x0018	RW	Comparator Interrupt Enable Register	0x0000_0000	20.2.5
COMPIST	0x001C	RO	Comparator Interrupt Status Register	0x0000_0000	20.2.6
COMPICLR	0x0020	RW	Comparator Interrupt Clear Register	0x0000_0000	20.2.7

20.2.1 COMP0CR: Comparator 0 control register

COMP0CR=0x4000_3420

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	COMPREFEN	WINDEN	Reserved	COMP0_MODE	Reserved	COMP0_HYSSEL	Reserved	COMP0_EN	Reserved	COMP0_INNSEL	Reserved
-	0	0	-	0	-	0	-	0	-	0	-
-	RW	RW	-	RW	-	RW	-	RW	-	RW	-

25	COMPREFEN	Comparator internal reference voltage enable	
		0	Disable internal reference voltage
		1	Enable internal reference voltage
24	WINDEN	Window mode Enable	
		0	Disable Window mode
		1	Enable Window mode

NOTE: In Window mode, the input of Comparator 1 is connected to the input of comparator 0 (PA [01]), and the comparator 1 and PA [03] are disconnected.

21 20	COMP0_MODE	Comparator 0 mode Selection	
		00	UL power
		01	Low power
		10	Medium power
		11	High speed
17 16	COMP0_HYSSEL	Comparator 0 Hysteresis mode selection	
		00	Off
		01	Low
		10	Medium
		11	High
12	COMP0_EN	Comparator 0 Enable	
		0	Disable Comparator
		1	Enable Comparator
6 4	COMP0_INNSEL	Comparator 0 Reference(input -) Selection	
		000	PA[00]
		001	PA[04] (shard with DAC_OUT1)
		010	PA[05] (shard with DAC_OUT2)
		011	Vref 1.2V
		100	Vref 0.9V
		101	Vref 0.6V
		110	Vref 0.3V
		111	DAC_OUT3

NOTE:

1. Since DAC_OUT1 and DAC_OUT2 are connected to the port, when using DAC_OUT as a comparator reference, the GPIO function of the corresponding port cannot be used. Since DAC_OUT3 is not connected to the port, GPIO function is not affected.

20.2.2 COMP1CR: Comparator 1 control register

COMP1CR=0x4000_3424

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	COMP1_MODE	Reserved	COMP1_HYSSEL	Reserved	COMP1_EN	Reserved	COMP1_INNSEL	Reserved
-	0	-	0	-	0	-	0	-
-	RW	-	RW	-	RW	-	RW	-

21 20	COMP1_MODE	Comparator 1 mode Selection	
		00	UL power
		01	Low power
		10	Medium power
		11	High speed
17 16	COMP1_HYSSEL	Comparator 1 Hysteresis mode selection	
		00	Off
		01	Low
		10	Medium
		11	High
12	COMP1_EN	Comparator 1 Enable	
		0	Disable Comparator
		1	Enable Comparator
6 4	COMP1_INNSEL	Comparator 0 Reference(input -) Selection	
		000	PA[02]
		001	PA[04] (shard with DAC_OUT1)
		010	PA[05] (shard with DAC_OUT2)
		011	Vref 1.2V
		100	Vref 0.9V
		101	Vref 0.6V
		110	Vref 0.3V
		111	DAC_OUT3

NOTE:

1. Since DAC_OUT1 and DAC_OUT2 are connected to the port, when using DAC_OUT as a comparator reference, the GPIO function of the corresponding port cannot be used. Since DAC_OUT3 is not connected to the port, GPIO function is not affected.

20.2.3 COMPDBNC: Comparator debounce register

COMPDBNC=0x4000_3430

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
DBNCTB		Reserved	DBCSEL	C1DBNC	C0DBNC
0		-	0	0	0
RW		-	RW	RW	RW

31 16	DBNCTB	Debounce time base counter System clock/(DBNCTB *2) becomes shift clock of debounce logic When DBNCTB is 0, system clock would be debounce clock.
8	DBCSEL	Debounce time base clock selection 0 System clock 1 LSI750kHz
7 4	C1DBNC	Debounce shift Selection When it is 0x0, debounce function is disable Shift number of debounce logic is (C1DBNC + 1) when C1DBNC is more than 1
3 0	C0DBNC	Debounce shift Selection When it is 0x0, debounce function is disable Shift number of debounce logic is (C0DBNC + 1) when C0DBNC is more than 1

20.2.4 COMPICON: Comparator Interrupt control register

COMPICON=0x4000_3434

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			TPOL1	TPOL0	Reserved	IPOL1	IPOL0	Reserved	C1IMODE	C0IMODE
-			0	0	-	0	0	-	0	0
-			RW	RW	-	RW	RW	-	RW	RW

13	TPOL1	Comparator Trigger output polarity(to trigger other IP) 0 output normal (comparator out high activates trigger) 1 output inverted (XOR)
12	TPOL0	Comparator Trigger output polarity(to trigger other IP) 0 output normal (comparator out high activates trigger) 1 output inverted (XOR)

9	IPOL1	Comparator 1 interrupt polarity(level mode)	
		0	interrupt at comparator out high
		1	interrupt at comparator out low
8	IPOL0	Comparator 0 interrupt polarity(level mode)	
		0	interrupt at comparator out high
		1	interrupt at comparator out low
3 2	C1IMODE	Comparator 1 Interrupt Flag bit.	
		00	level interrupt (by IPOL1)
		01	rising edge interrupt
		10	falling edge interrupt
		11	both edge interrupt
1 0	C0IMODE	Comparator 0 Interrupt Flag bit.	
		00	level interrupt (by IPOL0)
		01	rising edge interrupt
		10	falling edge interrupt
		11	both edge interrupt

20.2.5 COMPIEN: Comparator interrupt enable register

COMPIEN=0x4000_3438

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				C1IEN	C0IEN
-				0	0
-				RW	RW

1	C1IEN	Comparator 1 Enable	
		0	Disable
		1	Enable
0	C0IEN	Comparator 0 Enable	
		0	Disable
		1	Enable

20.2.6 COMPIST: Comparator interrupt status register

COMPIST=0x4000_343C

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				C1IRQ	C0IRQ
-				0	0
-				RO	RO

1	C1IRQ	Comparator 1 interrupt Status	
		0	No Comparator Interrupt
		1	Comparator Interrupt asserted
0	C0IRQ	Comparator 0 interrupt Status	
		0	No Comparator Interrupt
		1	Comparator Interrupt asserted

20.2.7 COMPICLR: Comparator interrupt clear register

COMPICLR=0x4000_3440

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				C1ICLR	C0ICLR
-				0	0
-				RW	RW

1	C1ICLR	Comparator 1 Interrupt Clear (write "1" to clear C1IRQ)
0	C0ICLR	Comparator 0 Interrupt Clear (write "1" to clear C0IRQ)

21. Cyclic redundancy check and checksum (CRC checksum)

Cyclic redundancy check (CRC) generator is used to get a 16-bit CRC code from Flash ROM and a generator polynomial. Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. In the scope of the functional safety standards, they offer a means of verifying the Flash memory integrity. CRC generator helps compute a signature of the software during runtime, to be compared with a reference signature.

CRC generator of A31G32x series features the followings:

- Auto CRC (DMA) and User CRC Mode.
- Polynomial:
  - CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
  - CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
- CRC Mode and Checksum Mode.

21.1 CRC and checksum block diagram

Figure 150 describes the CRC and checksum in a block diagram.

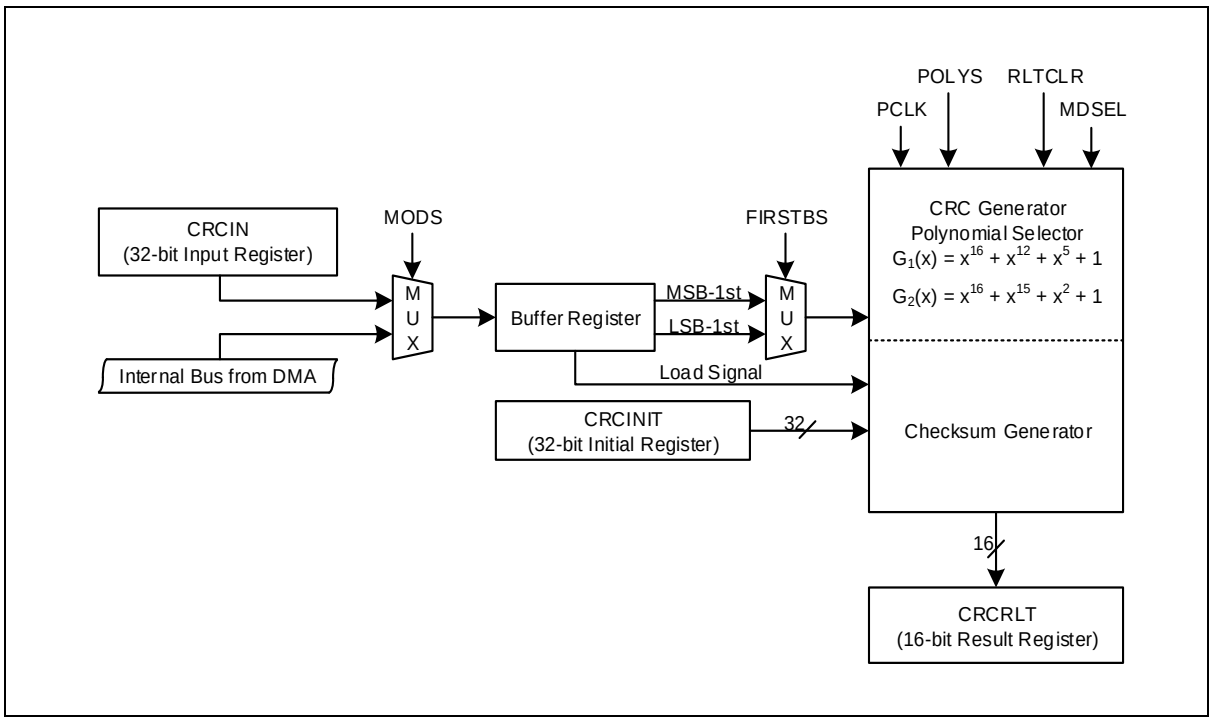


Figure 150. CRC and Checksum Block Diagram

21.2 Registers

Base address of the CRC and checksum block is introduced in the followings:

Table 83. Base Address of CRC

Name	Base address
CRC	0x4000_0300

Table 84. CRC Register Map

Name	Offset	Type	Description	Reset value	Reference
CRC_CR	0x0000	RW	CRC/Checksum Control Register	0x0000_0000	21.2.1
CRC_IN	0x0004	RW	CRC/Checksum Input Data Register	0x0000_0000	21.2.2
CRC_RLT	0x0008	RO	CRC/Checksum Result Data Register	0x0000_FFFF	21.2.3
CRC_INIT	0x000C	RW	CRC/Checksum Initial Data Register	0x0000_0000	21.2.4

21.2.1 CRC_CR: CRC control register

CRC_CR is a 32-bit register, and able to do 32/16/8-bit access.

CRC_CR=0x4000_0300

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			CRCINTEN	CRCINTF	Reserved	RLTCLR	MDSEL	POLYS	Reserved	FIRSTBS	CRCRUN
-			0	0	-	0	0	0	-	0	0
-			RW	RW	-	RW	RW	RW	-	RW	RW

9	CRCINTEN	CRC DMA interrupt enable bit	
		0	Disable
		1	Enable
8	CRCINTF	CRC DMA done interrupt flag bit	
		0	No request occurred
		1	Request occurred, This bit is cleared to '0' when write '1'.

6	RLTCLR	CRC/Checksum Result Data Register (CRC_RLT) Initialization bit.	
		0	No effect.
		1	Initialize the CRC_RLT register with the value of CRC_INIT (This bit is automatically cleared to "0b" after operation)
NOTE: The checksum is calculated by byte unit. Ex) On 0x34A7E991, CRCRLT = 0x34 + 0xA7 + 0xE9 + 0x91.			
5	MDSEL	CRC/Checksum Selection bit.	
		0	Select CRC.
		1	Select checksum.
		Note) The checksum is calculated by byte unit. Ex) On 0x34A7E991, CRCRLT = 0x34 + 0xA7 + 0xE9 + 0x91.	
4	POLYS	Polynomial Selection bit. (CRC only)	
		0	Select CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
		1	Select CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
1	FIRSTBS	First Shifted-in Selection bit. (CRC only)	
		0	MSB-1st.
		1	LSB-1st.
0	CRCRUN	CRC/Checksum enable control. In DMA mode, this bit is a busy bit.	
		0	CRC/Checksum disabled.
		1	CRC/Checksum enabled. In DMA mode, this bit is automatically cleared to "0b" after CRC operation

NOTES:

1. The CRCRUN bit should be set to "1b" last time after setting appropriate values to the registers.
2. The 4 "NOP instruction" should follow immediately after this bit is set.
3. It will be calculated every writing data to the CRC_IN register during CRCRUN=1.
4. It is prohibited writing any data to the CRC_IN register during CRCRUN=0 or running in DMA mode.

21.2.2 CRC_IN: CRC input data register

CRC_IN is a 32-bit register.

CRC_IN=0x4000_0304

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
INDATA			
0x00000000 RW			

31 0	INDATA	CRC Input Data bits.
NOTE: The CRC_IN register should be written by 1-word (32-bits).		

21.2.3 CRC_RLT: CRC result data register

CRC_RLT is a 32-bit register, and able to do 32/16/8-bit access.

CRC_RLT=0x4000_0308

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		RLTDATA	
-		0xFFFF	
-		RW	

15 0	RLTDATA	CRC Result Data bits.

21.2.4 CRC_INIT: CRC initial data register

CRC_INIT is a 32-bit register, and able to do 32/16/8-bit access.

CRC_INIT=0x4000_030C

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		INIDATA	
-		0x0000	
-		RW	

15 0	INIDATA	CRC Initial Data bits.

21.3 Functional description

21.3.1 CRC polynomial structure

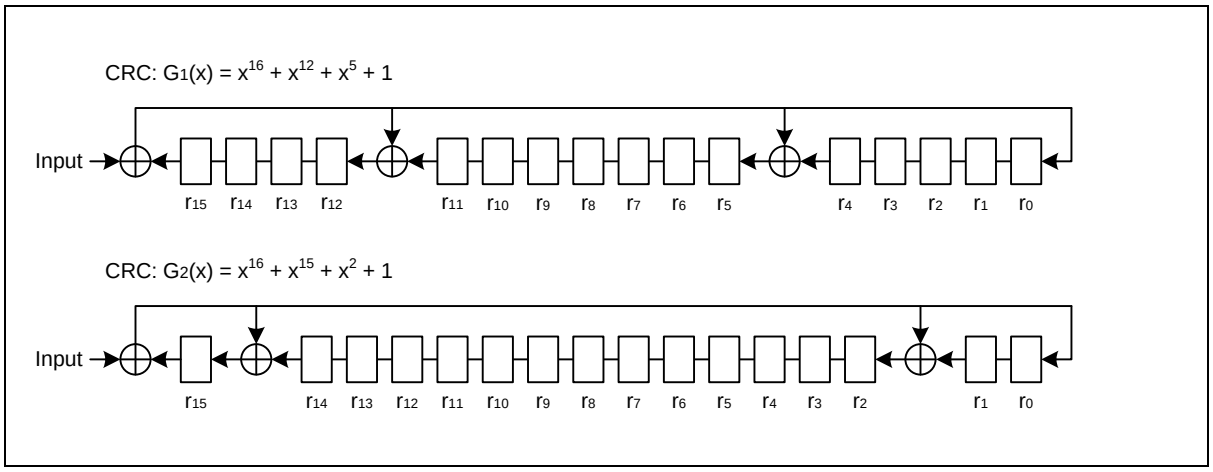


Figure 151. CRC Polynomial Structure

21.3.2 CRC operation procedure in DMA mode

1. CRC/Checksum Enable and Clock Enable. (SCU_PCER, SCU_PER)
2. Set CRC initial data register. (CRCINIT)
3. Set CRC control register.
4. CRC operation enable (CRCRUN = 1)
5. DMA configuration & operation
6. Read the CRC result.

21.3.3 CRC operation procedure in user CRC and checksum mode

1. CRC/Checksum Enable and Clock Enable (SCU_PCER, SCU_PER)
2. Set CRC initial data register. (CRC_INIT)
3. Set CRC control register.
4. CRC operation enable. (CRCRUN = 1)
5. Input CRC Data at CRC_IN.
6. CRC stop and read CRC result.

22. USB full speed device interface

USB block of A31G24x series controls USB 2.0 full speed interface. The USB block features the followings:

- Support for the following speeds:
  - Full-Speed (FS, 12-Mbps)
  - Low-Speed (LS, 1.5-Mbps)
- USB 2.0 full-speed
- Configurable number of endpoints from 1 to 4
- 2 KB dynamic FIFO support.
- Control/bulk/interrupt transfer support.

22.1 USB block diagram

Figure 152 describes the USB interface block of A31G32x series in a block diagram.

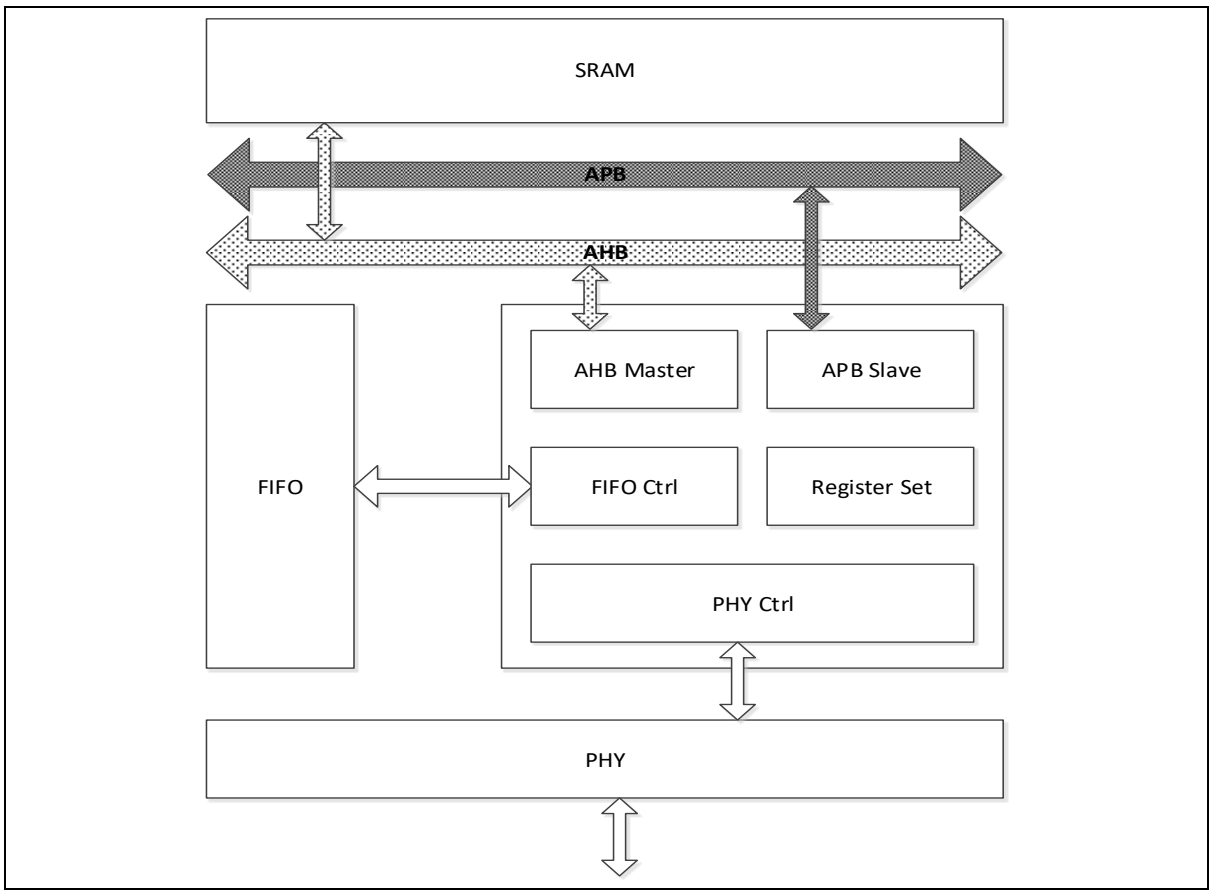


Figure 152. USB Interface Block Diagram

22.2 Registers

Base address of the USB interface block is introduced in the followings:

Table 85. Base Address of USB Interface

Name	Base address
USB	0x5000_0000

Table 86. USB Interface Register Map

Name	Offset	Type	Description	Reset value	Reference
GOTGCTL	0x0000	RW	Control and Status Register	0x0089_0000	22.2.1
GOTGINT	0x0004	RW	Interrupt Register	0x0010_0000	22.2.2
GAHBCFG	0x0008	RW	AHB Configuration Register	0x0000_0000	22.2.3
GUSBCFG	0x000C	RW	USB Configuration Register	0x0000_1448	22.2.4
GRSTCTL	0x0010	RW	Reset Register	0x8000_0000	22.2.5
GINTSTS	0x0014	RW	Interrupt Register	0x5400_0034	22.2.6
GINTMSK	0x0018	RW	Interrupt Mask Register	0x0000_0000	22.2.7
GRXSTSR	0x001C	RW	Receive Status Debug Read Register	0x0000_0000	22.2.8
GRXSTSP	0x0020	RW	Receive Status Read/Pop Register	0x0000_0000	22.2.9
GRXFSIZ	0x0024	RW	Receive FIFO Size Register	0x0000_0200	22.2.10
GNPTXFSIZ	0x0028	RW	Non-periodic Transmit FIFO Size Register	0x0040_0200	22.2.11
GSNPSID	0x0040	RW	Synopsys ID Register	0x5531_100A	22.2.12
GHWCFG1	0x0044	RW	User HW Config1 Register	0x0000_0000	22.2.13
GHWCFG2	0x0048	RW	User HW Config2 Register	0x0288_1114	22.2.14
GHWCFG3	0x004C	RW	User Hardware Config3 Register	0x01D8_4468	22.2.15
GHWCFG4	0x0050	RW	User Hardware Config4 Register	0XD200_4030	22.2.16
GLPMCFG	0x0054	RW	LPM Config Register	0x0000_0000	22.2.17
GPWRDN	0x0058	RW	Global Power Down register	0x0000_0000	22.2.18
GDFIFOCFG	0x005C	RW	Global DFIFO Configuration Register	0x01D8_0200	22.2.19

Table 86. USB Interface Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
DIEPTXFi	0x013C	RW	Device IN Endpoint Transmit FIFO Size Register	0x0000_0000	22.2.20
DCFG	0x0800	RW	Device Configuration Register	0x0810_0000	22.2.21
DCTL	0x0804	RW	Device Control Register	0x0000_0002	22.2.22
DSTS	0x0808	RW	Device Status Register	0x0000_0002	22.2.23
DIEPMSK	0x0810	RW	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000	22.2.24
DOEPMSK	0x0814	RW	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000	22.2.25
DAINT	0x0818	RW	Device All Endpoints Interrupt Register	0x0000_0000	22.2.26
DAINTMSK	0x081C	RW	Device All Endpoints Interrupt Mask Register	0x0000_0000	22.2.27
DTHRCTL	0x0830	RW	Device Threshold Control Register	0x0C10_0020	22.2.28
DIEPEMPMS	0x0834	RW	Device IN Endpoint FIFO Empty Interrupt Mask Register	0x0000_0000	22.2.29
DIEPCTL0	0x0900	RW	Device Control IN Endpoint 0 Control Register	0x0000_8000	22.2.30
DIEPINT0	0x0908	RW	Device IN Endpoint 0 Interrupt Register	0x0000_0080	22.2.31
DIEPDMA0	0x0914	RW	Device IN Endpoint 0 DMA Address Register	0x0000_0000	22.2.32
DTXFSTS0	0x0918	RW	Device IN Endpoint Transmit FIFO Status Register	0x0000_0040	22.2.33
DIEPDMAB0	0x091C	RW	Device IN Endpoint 16 Buffer Address Register	0x0000_0000	22.2.34
DIEPCTLi	0x0920	RW	Device Control IN Endpoint I Control Register	0x0000_0000	22.2.35
DIEPINTi	0x0928	RW	Device IN Endpoint I Interrupt Register	0x0000_0080	22.2.36
DIEPDMAi	0x0934	RW	Device IN Endpoint I DMA Address Register	0x0000_0000	22.2.37

Table 86. USB Interface Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
DTXFSTSi	0x0938	RW	Device IN Endpoint Transmit FIFO Status Register	0x0000_0040	22.2.38
DIEPDMABi	0x093C	RW	Device IN Endpoint I Buffer Address Register	0x0000_0000	22.2.39
DOEPCTL0	0x0B00	RW	Device Control OUT Endpoint 0 Control Register	0x0000_8000	22.2.40
DOEPINT0	0x0B08	RW	Device OUT Endpoint 0 Interrupt Register	0x0000_0000	22.2.41
DOEPDMA0	0x0B14	RW	Device OUT Endpoint 0 DMA Address Register	0x0000_0000	22.2.42
DOEPDMAB	0x0B1C	RW	Device OUT Endpoint 16 Buffer Address Register	0x0000_0000	22.2.43
DOEPCTLi	0x0B20	RW	Device Control OUT Endpoint I Control Register	0x0000_0000	22.2.44
DOEPINTi	0x0B28	RW	Device OUT Endpoint I Interrupt Register	0x0000_0000	22.2.45
DOEPDMAi	0x0B34	RW	Device OUT Endpoint I DMA Address Register	0x0000_0000	22.2.46
DOEPDMAB	0x0B3C	RW	Device OUT Endpoint I Buffer Address Register	0x0000_0000	22.2.47
PCGCCTL	0x0E00	RW	Power and Clock Gating Control Register	0x0000_0000	22.2.48

22.2.1 GOTGCTL: Control and status register

GOTGCTL register controls behavior of the controller, and reflects its status.

GOTGCTL =0x5000_0000

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	MultValIdBC	Reserved	
-	0x02	-	
-	R	-	

26 22	MultValIdBC	Multi Valued ID pin (MultValIdBC) Battery Charger ACA inputs in the following order: • Bit 26: rid_float. • Bit 25: rid_gnd • Bit 24: rid_a • Bit 23: rid_b • Bit 22: rid_c Values: • 0x10 (RID_FLOAT): B-Device not connected to ACA • 0x8 (RID_GND): A-Device not connected to ACA • 0x4 (RID_A): A-Device connected to ACA • 0x2 (RID_B): B-Device connected to ACA. VBUS is off. • 0x1 (RID_C): B-Device connected to ACA. VBUS is on. Value After Reset: Varies based on Configuration Exists: OTG_BC_SUPPORT==1 Volatile: true

22.2.2 GOTGINT: Interrupt register

Application reads this register whenever an interrupt occurs. Corresponding bits in this register is cleared after the reading.

GOTGINT =0x5000_0004

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		MultValIpChn		Reserved	
-		1		-	
-		RW		-	

20	MultValIpChng	This bit when set indicates that there is a change in the value of at least one ACA pin value. Values: • 0x1 (ACA_PIN_CHANGE): Indicates there is a change in ACA pin value. • 0x0 (NO_ACA_PIN_CHANGE): Indicates there is no change in ACA pin value.

22.2.3 GAHBCFG: AHB configuration register

This register can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

GAHBCFG =0x5000_0008

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	InvDescEndianes	AHBSingle	NotiAllDmaWrit	RemMemSupp	Reserved		DMAEn	HbstLen	GlblIntrMsk
-	0	0	0	0	-		0	0	0
-	RW	RW	RW	RW	-		RW	RW	RW

24	InvDescEndianes	Invert Descriptor Endianess (InvDescEndianess) • 1'b0: Descriptor Endianness is same as AHB Master Endianness. • 1'b1: -If the AHB Master endianness is Big Endian, the Descriptor Endianness is Little Endian. -If the AHB Master endianness is Little Endian, the Descriptor Endianness is Big Endian. Values: • 0x0 (DISABLE): Descriptor Endianness is same as AHB Master Endianness • 0x1 (ENABLE): Descriptor Endianness is opposite to AHB Master Endianness Value After Reset: 0x0 Exists: OTG_EN_DESC_DMA == 1 Volatile: true
23	AHBSingle	AHB Single Support (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the core is operating in DMA mode. • 1'b0: The remaining data in the transfer is sent using INCR burst size. • 1'b1: The remaining data in the transfer is sent using Single burst size. Note: If this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus). Values: • 0x0 (INCRBURST): The remaining data in the transfer is sent using INCR burst size • 0x1 (SINGLEBURST): The remaining data in the transfer is sent using Single burst size Value After Reset: 0x0 Exists: Always

22	NotiAllDmaWrit	Notify All Dma Write Transactions (NotiAllDmaWrit) This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. • GAHBCFG.NotiAllDmaWrit = 1 The core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. • GAHBCFG.NotiAllDmaWrit = 0 The core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint. Values: • 0x1 (ALLTRANS): The core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint • 0x0 (LASTTRANS): Value After Reset: 0x0 Exists: Always Volatile: true
21	RemMemSupp	Remote Memory Support (RemMemSupp) This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. • GAHBCFG.RemMemSupp=1 The int_dma_req output signal is asserted when the DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from the controller. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. • GAHBCFG.RemMemSupp=0 The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the Core Boundary and it does not wait for the sys_dma_done signal to complete the DATA transfers. Values: • 0x0 (DISABLED): Remote Memory Support Feature disabled • 0x1 (ENABLED): Remote Memory Support Feature enabled Value After Reset: 0x0 Exists: Always Volatile: true
5	DMAEn	DMA Enable (DMAEn) Reset: 1'b0 Values: • 0x1 (DMAMODE): Core operates in a DMA mode Value After Reset: 0x0 Exists: OTG_ARCHITECTURE != 0 Volatile: true

4 1	HbstLen	Burst Length/Type (HbstLen) Internal DMA Mode AHB Master burst type: • 4'b0000 Single • 4'b0001 INCR • 4'b0011 INCR4 • 4'b0101 INCR8 • 4'b0111 INCR16 • Others: Reserved Values: • 0x1 (WORD4ORINCR): 4 words or INCR • 0x5 (WORD64ORINCR8): 64 words or INCR8 • 0x3 (WORD16ORINCR4): 16 words or INCR4 • 0x0 (WORD1ORSINGLE): 1 word or single • 0x7 (WORD256ORINCR16): 256 words or INCR16 Value After Reset: 0x0 Exists: Always Volatile: true
0	GlblIntrMsk	Global Interrupt Mask (GlblIntrMsk) The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the controller. • 1'b0: Mask the interrupt assertion to the application. • 1'b1: Unmask the interrupt assertion to the application. Values: • 0x0 (MASK): Mask the interrupt assertion to the application. • 0x1 (NOMASK): Unmask the interrupt assertion to the application. Value After Reset: 0x0 Exists: Always

22.2.4 GUSBCFG: USB configuration register

This register can be used to configure the core after power-on. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

GUSBCFG =0x5000_000C

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
CorruptTxPkt	Reserved				PHYSel	FSIntf	Reserved	ToutCal
0	-				1	0	-	0
W	-				RW	RW	-	RW

31	CorruptTxPkt	Corrupt Tx packet (CorruptTxPkt) This bit is for debug purposes only. Never Set this bit to 1. The application should always write 1'b0 to this bit. Values: • 0x1 (DEBUG): Debug Mode • 0x0 (NODEBUG): Normal Mode

6	PHYSel	PHYSel USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) • 1'b1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected in, this bit is always 0, with Write Only access. Values: • 0x1 (USB11): USB 1.1 full-speed serial transceiver is selected
5	FSIntf	Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. • 1'b0: 6-pin unidirectional full-speed serial interface • 1'b1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Write Only access. If a USB 1.1 FS interface was selected, Then the application can Set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write. Values: • 0x1 (FS3PIN): 3-pin bidirectional full-speed serial interface • 0x0 (FS6PIN): 6-pin unidirectional full-speed serial interface
2 0	ToutCal	FS Timeout Calibration (ToutCal) The number of PHY clocks that the application programs in this field is added to the full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are as follows: Full-speed operation: • One 48-MHz PHY clock = 0.25 bit times

22.2.5 GRSTCTL: Reset register

The application uses this register to reset various hardware features inside the controller.

GRSTCTL =0x5000_0010

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
AHBIdle	DMAReq	Reserved		TxFNum		TxFFlsh	RxFFlsh	Reserved	PIUFSSftRst	CSftRst
1	0	-		0		0	0	-	0	0
R	R	-		RW		RW	RW	-	RW	RW

31	AHBIdle	AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition. Values: • 0x1 (ACTIVE): AHB Master Idle • 0x0 (INACTIVE): Not Idle Value After Reset: 0x1

30	DMAReq	DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug. Values: • 0x1 (ACTIVE): DMA request is in progress • 0x0 (INACTIVE): No DMA request Value After Reset: 0x0
10 6	TxFNum	TxFIFO Number (TxFNum) This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. • 5'h0: -Tx FIFO 0 flush in device mode when in dedicated FIFO mode. • 5'h1: -TXFIFO 1 flush in device mode when in dedicated FIFO mode. • 5'h2: -TXFIFO 2 flush in device mode when in dedicated FIFO mode. ... • 5'h4 -TXFIFO 4 flush in device mode when in dedicated FIFO mode. • 5'h10: Flush all the transmit FIFOs. Values: • 0x3 (TXF3): -Periodic TxFIFO 3 flush in device mode when in shared FIFO operation –TXFIFO 3 flush in device mode when in dedicated FIFO mode • 0x4 (TXF4): -Periodic TxFIFO 4 flush in device mode when in shared FIFO operation –TXFIFO 4 flush in device mode when in dedicated FIFO mode • 0x2 (TXF2): -Periodic TxFIFO 2 flush in device mode when in shared FIFO operation –TXFIFO 2 flush in device mode when in dedicated FIFO mode • 0x1 (TXF1): -Periodic TxFIFO 1 flush in device mode when in shared FIFO operation –TXFIFO 1 flush in device mode when in dedicated FIFO mode • 0x0 (TXF0): -Periodic TxFIF O flush in device mode when in shared FIFO operation –TXFIFO 0 flush in device mode when in dedicated FIFO mode Value After Reset: 0x0
5	TxFFlsh	TxFIFO Flush (TxFFlsh) This bit selectively flushes a single or all transmit FIFOs, but cannot do so If the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: • ReadNAK Effective Interrupt ensures the core is not reading from the FIFO • WriteGRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk. Values: • 0x1 (ACTIVE): Selectively flushes a single or all transmit FIFOs • 0x0 (INACTIVE): No Flush Value After Reset: 0x0
4	RxFFlsh	RxFIFO Flush (RxFFlsh) The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the controller is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires eight clocks (slowest of PHY or AHB clock) to clear. Values: • 0x1 (ACTIVE): Flushes the entire RxFIFO • 0x0 (INACTIVE): Does not flush the entire RxFIFO Value After Reset: 0x0
1	PIUFSSftRst	PIU FS Dedicated Controller Soft Reset (PIUFSSftRst)

		Resets the PIU FS Dedicated Controller All module state machines in FS Dedicated Controller of PIU are reset to the IDLE state. Used to reset the FS Dedicated controller in PIU in case of any PHY Errors like Loss of activity or Babble Error resulting in the PHY remaining in RX state for more than one frame boundary. This is a self-clearing bit and core clears this bit after all the necessary logic is reset in the core. Values: • 0x1 (RESET_ACTIVE): PIU FS Dedicated Controller Soft Reset • 0x0 (RESET_INACTIVE): No Reset Value After Reset: 0x0
0	CSftRst	Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: • Clears the interrupts and all the CSR registers except the following register bits: -PCGCCTL.RstPdwnModule -PCGCCTL.GateHclk -PCGCCTL.PwrClmp -PCGCCTL.StopPPhyLPwrClkSelclk -GUSBCFG.PhyLPwrClkSel -GUSBCFG.FSIntf -GGPIO -GPWRDN -HCFG.FSLSPclkSel -DCFG.DevSpd -DCTL.SftDiscon • All module state machines • All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. • Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. • When Hibernation feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation. Values: • 0x1 (ACTIVE): Resets hclk and phy_clock domains • 0x0 (NOTACTIVE): No reset Value After Reset: 0x0

22.2.6 GINTSTS: Interrupt register

This register interrupts the application for system-level events in the current mode. To clear the interrupt status bits of type R_SS_WC, the application must write 1'b1 into the bit.

The FIFO status interrupts are read only; once software reads from or writes to the FIFO while servicing these interrupts, FIFO interrupt conditions are cleared automatically.

The application must clear the GINTSTS register at initialization before unmasking the interrupt bit to avoid any interrupts generated prior to initialization.

GINTSTS =0x5000_0014

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
WkUpInt	Reserved	LPM_Int	Reserved	ResetDet	FetSusp	Reserved	OEPInt	IEPInt	EPMis	RstrDoneInt	EOPF	ISOOutDrop	EnumDone	USBRst	USBSusp	ErlySusp	Reserved	GOUTNakEff	GINNakEff	Reserved	RxFLvl	Sof	Reserved
0	-	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	-	0	0	-	1	0	-
RW	-	RW	-	RW	RW	-	R	R	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	R	-	R	RW	-

31	WkUpInt	Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. • During Suspend(L2): -This interrupt is asserted only when Host Initiated Resume is detected on USB. For more information, see 'Partial Power-Down and Clock Gating Programming Model' in the Programming Guide. • During LPM(L1): -This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. For more information, see 'LPM Entry and Exit Programming Model' in the Programming Guide. Values: • 0x1 (ACTIVE): Resume or Remote Wakeup Detected Interrupt • 0x0 (INACTIVE): Not active
27	LPM_Int	LPM Transaction Received Interrupt (LPM_Int). This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. Has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). Values: • 0x1 (ACTIVE): LPM Transaction Received Interrupt • 0x0 (INACTIVE): Not Active
23	ResetDet	Reset detected Interrupt (ResetDet) In Device mode, this interrupt is asserted when a reset is detected on the USB in partial power-down mode when the device is in Suspend. Values: • 0x1 (ACTIVE): Reset detected Interrupt • 0x0 (INACTIVE): Not active

22	FetSusp	Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: • Sets a Global non-periodic IN NAK handshake • Disables In endpoints • Flushes the FIFO • Determines the token sequence from the IN Token Sequence Learning Queue • Re-enables the endpoints • Clears the Global non-periodic IN NAK handshake If the Global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received. The core generates an 'IN token received when FIFO empty' interrupt. The OTG Then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a Global NAK handshake. Alternatively, the application can mask the IN token received when FIFO empty interrupt when clearing a Global IN NAK handshake. Values: • 0x1 (ACTIVE): Data Fetch Suspended • 0x0 (INACTIVE): Not active
19	OEPInt	OUT Endpoints Interrupt (OEPInt) The controller sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit. Values: • 0x1 (ACTIVE): OUT Endpoints Interrupt • 0x0 (INACTIVE): Not active
18	IEPInt	IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit. Values: • 0x1 (ACTIVE): IN Endpoints Interrupt • 0x0 (INACTIVE): Not active
17	EPMis	Endpoint Mismatch Interrupt (EPMis) Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired. Values: • 0x1 (ACTIVE): Endpoint Mismatch Interrupt • 0x0 (INACTIVE): Not active

16	RstrDoneInt	Restore Done Interrupt (RstrDoneInt) The controller sets this bit to indicate that the restore command after Hibernation was completed by the core. The controller continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field. Values: • 0x1 (ACTIVE): Restore Done Interrupt • 0x0 (INACTIVE): Not active
15	EOPF	End of Periodic Frame Interrupt (EOPF) Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe. Values: • 0x1 (ACTIVE): End of Periodic Frame Interrupt • 0x0 (INACTIVE): Not active
14	ISOOutDrop	Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The controller sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint. Values: • 0x1 (ACTIVE): Isochronous OUT Packet Dropped Interrupt • 0x0 (INACTIVE): Not active
13	EnumDone	Mode: Device only Enumeration Done (EnumDone) The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed. Values: • 0x1 (ACTIVE): Enumeration Done • 0x0 (INACTIVE): Not active
12	USBRst	USB Reset (USBRst) The controller sets this bit to indicate that a reset is detected on the USB. Values: • 0x1 (ACTIVE): USB Reset • 0x0 (INACTIVE): Not active
11	USBSusp	USB Suspend (USBSusp) The controller sets this bit to indicate that a suspend was detected on the USB. The controller enters the Suspended state when there is no activity on the linestate signal for an extended period of time. Values: • 0x1 (ACTIVE): USB Suspend • 0x0 (INACTIVE): Not Active
10	ErlySusp	Early Suspend (ErlySusp) The controller sets this bit to indicate that an Idle state has been detected on the USB for 3 ms. Values: • 0x1 (ACTIVE): 3ms of Idle state detected • 0x0 (INACTIVE): No Idle state detected
7	GOUTNakEff	Global OUT NAK Effective (GOUTNakEff) Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), Set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak). Values: • 0x1 (ACTIVE): Global OUT NAK Effective • 0x0 (INACTIVE): Not Active

6	GINNakEff	Global IN Non-periodic NAK Effective (GINNakEff) Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak) set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit Set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit. Values: • 0x1 (ACTIVE): Set Global Non-periodic IN NAK bit • 0x0 (INACTIVE): Global Non-periodic IN NAK not active
4	RxFLvl	RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO. Values: • 0x1 (ACTIVE): Rx Fifo is not empty • 0x0 (INACTIVE): Rx Fifo is empty
3	Sof	Start of (micro)Frame (Sof) In Device mode, the controller sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro) Frame number. This interrupt is seen only when the core is operating at FS. This bit can be set only by the core and the application must write 1 to clear it. Note: This register may return 1'b1 if read immediately after power-on reset. If the register bit reads 1'b1 immediately after power-on reset, it does not indicate that an SOF has been received. The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset the application can clear the bit. Values: • 0x1 (ACTIVE): Start of Frame • 0x0 (INTACTIVE): No Start of Frame

22.2.7 GINTMSK: Interrupt mask register

This register works with the Interrupt Register (GINTSTS) to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the GINTSTS register bit corresponding to that interrupt is still set.

GINTMSK =0x5000_0018

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
WkUpIntMsk	Reserved			LPM_IntMsk	Reserved			ResetDetMsk	FetSuspMsk	Reserved		OEPIntMsk	IEPIntMsk	EPMisMsk	RstrDoneIntMsk	EOPFMsk	ISOOutDropMsk	EnumDoneMsk	USBRstMsk	USBSuspMsk	ErlySuspMsk	Reserved		GOUTNakEffMsk	GINNakEffMsk	Reserved	RxFLvlMsk	SofMsk	OTGIntMsk	Reserved	
0	-			0	-			0	0	-		0	0	0	0	0	0	0	0	0	0	-		0	0	-	0	0	0	-	
RW	-			RW	-			RW	RW	-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-		RW	RW	-	RW	RW	RW	-	

31	WkUpIntMsk	Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk) The WakeUp bit is used for LPM state wake up in a way similar to that of wake up in suspend state. Values: • 0x0 (MASK): Resume or Remote Wakeup Detected Interrupt Mask • 0x1 (NOMASK): Unmask Resume Remote Wakeup Detected Interrupt Value After Reset: 0x0
27	LPM_IntMsk	LPM Transaction Received Interrupt (LPM_Int) LPM Transaction received interrupt Mask Values: • 0x0 (MASK): LPM Transaction received interrupt Mask • 0x1 (NOMASK): No LPM Transaction received interrupt Mask Value After Reset: 0x0
23	ResetDetMsk	Reset detected Interrupt Mask (ResetDetMsk) Values: • 0x0 (MASK): Reset detected Interrupt Mask • 0x1 (NOMASK): No Reset detected Interrupt Mask Value After Reset: 0x0
22	FetSuspMsk	Data Fetch Suspended Mask (FetSuspMsk) Values: • 0x0 (MASK): Data Fetch Suspended Mask • 0x1 (NOMASK): No Data Fetch Suspended Mask Value After Reset: 0x0
19	OEPIntMsk	Mode: Device only OUT Endpoints Interrupt Mask (OEPIntMsk) Values: • 0x0 (MASK): OUT Endpoints Interrupt Mask • 0x1 (NOMASK): No OUT Endpoints Interrupt Mask Value After Reset: 0x0
18	IEPIntMsk	Mode: Device only IN Endpoints Interrupt Mask (IEPIntMsk) Values: • 0x0 (MASK): IN Endpoints Interrupt Mask • 0x1 (NOMASK): No IN Endpoints Interrupt Mask Value After Reset: 0x0
17	EPMisMsk	Mode: Device only Endpoint Mismatch Interrupt Mask (EPMisMsk) Values: • 0x0 (MASK): Endpoint Mismatch Interrupt Mask • 0x1 (NOMASK): No Endpoint Mismatch Interrupt Mask Value After Reset: 0x0
16	RstrDoneIntMsk	Restore Done Interrupt Mask (RstrDoneIntMsk) Values: • 0x0 (MASK): Restore Done Interrupt Mask • 0x1 (NOMASK): No Restore Done Interrupt Mask Value After Reset: 0x0
15	EOPFMsk	Mode: Device only End of Periodic Frame Interrupt Mask (EOPFMsk) Values: • 0x0 (MASK): End of Periodic Frame Interrupt Mask • 0x1 (NOMASK): No End of Periodic Frame Interrupt Mask Value After Reset: 0x0
14	ISOOutDropMsk	Mode: Device only Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk) Values: • 0x0 (MASK): Isochronous OUT Packet Dropped Interrupt Mask • 0x1 (NOMASK): No Isochronous OUT Packet Dropped Interrupt Mask Value After Reset: 0x0

13	EnumDoneMsk	Mode: Device only Enumeration Done Mask (EnumDoneMsk) Values: • 0x0 (MASK): Enumeration Done Mask • 0x1 (NOMASK): No Enumeration Done Mask Value After Reset: 0x0
12	USBRstMsk	Mode: Device only USB Reset Mask (USBRstMsk) Values: • 0x0 (MASK): USB Reset Mask • 0x1 (NOMASK): No USB Reset Mask Value After Reset: 0x0
11	USBSuspMsk	Mode: Device only USB Suspend Mask (USBSuspMsk) Values: • 0x0 (MASK): USB Suspend Mask • 0x1 (NOMASK): No USB Suspend Mask Value After Reset: 0x0
10	ErlySuspMsk	Mode: Device only Early Suspend Mask (ErlySuspMsk) Values: • 0x0 (MASK): Early Suspend Mask • 0x1 (NOMASK): No Early Suspend Mask Value After Reset: 0x0
7	GOUTNakEffMsk	Mode: Device only Global OUT NAK Effective Mask (GOUTNakEffMsk) Values: • 0x0 (MASK): Global OUT NAK Effective Mask • 0x1 (NOMASK): No Global OUT NAK Effective Mask Value After Reset: 0x0
6	GINNakEffMsk	Mode: Device only, Global Non-periodic IN NAK Effective Mask (GINNakEffMsk) Values: • 0x0 (MASK): Global Non-periodic IN NAK Effective Mask • 0x1 (NOMASK): No Global Non-periodic IN NAK Effective Mask Value After Reset: 0x0
4	RxFLvlMsk	Receive FIFO Non-Empty Mask (RxFLvlMsk) Values: • 0x0 (MASK): Receive FIFO Non-Empty Mask • 0x1 (NOMASK): No Receive FIFO Non-Empty Mask Value After Reset: 0x0
3	SofMsk	Start of (micro)Frame Mask (SofMsk) Values: • 0x0 (MASK): Start of Frame Mask • 0x1 (NOMASK): No Start of Frame Mask Value After Reset: 0x0
2	OTGIntMsk	OTG Interrupt Mask (OTGIntMsk) Values: • 0x0 (MASK): OTG Interrupt Mask • 0x1 (NOMASK): No OTG Interrupt Mask Value After Reset: 0x0

22.2.8 GRXSTSR: Receive status debug read register

A read to GRXSTSR register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

NOTE: Do not read this register's reset value before configuring the core because the read value will be 'X' in the simulation.

GRXSTSR =0x5000_001C

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	FN	PktSts	DPID	BCnt	EPNum
-	0	0	0	0	0
-	R	R	R	R	R

24 21	FN	Mode: Device only Frame Number (FN) This is the least significant 4 bits of the (micro) Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported. Value After Reset: 0x0
20 17	PktSts	Packet Status (PktSts) Indicates the status of the received packet • 4'b0001: Global OUT NAK (triggers an interrupt) • 4'b0010: OUT data packet received • 4'b0011: OUT transfer completed (triggers an interrupt) • 4'b0100: SETUP transaction completed (triggers an interrupt) • 4'b0110: SETUP data packet received • Others: Reserved Reset:4'h0 Values: • 0x2 (INOUTDPRX): OUT data packet received • 0x1 (OUTNAK): Global OUT NAK (triggers an interrupt) • 0x4 (DSETUPCOM): SETUP transaction completed (triggers an interrupt) • 0x6 (DSETUPRX): SETUP data packet received • 0x3 (INOUTTRCOM): IN or OUT transfer completed (triggers an interrupt) Value After Reset: 0x0
16 15	DPID	Data PID (DPID) Indicates the Data PID of the received OUT data packet • 2'b00: DATA0 • 2'b10: DATA1 • 2'b01: DATA2 • 2'b11: MDATA Reset:2'h0 Values: • 0x0 (DATA0): DATA0 • 0x3 (MDATA): MDATA • 0x2 (DATA1): DATA1 • 0x1 (DATA2): DATA2 Value After Reset: 0x0

14 4	BCnt	Byte Count (BCnt) Indicates the byte count of the received data packet. Value After Reset: 0x0
3 0	EPNum	Endpoint Number (EPNum) Indicates the endpoint number to which the current received packet belongs. Value After Reset: 0x0

22.2.9 GRXSTSP: Receive status read/pop register

A read to the GRXSTSP register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO. The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

NOTE: Do not read this register's reset value before configuring the core because the read value is 'X' in the simulation.

GRXSTSR =0x5000_001C

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	7 6 5 4 3 2 1 0
Reserved	FN	PktSts	DPID	BCnt	EPNum
-	0	0	0	0	0
-	R	R	R	R	R

24 21	FN	Mode: Device only Frame Number (FN) This is the least significant 4 bits of the (micro) Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported. Value After Reset: 0x0
20 17	PktSts	Packet Status (PktSts) Indicates the status of the received packet • 4'b0001: Global OUT NAK (triggers an interrupt) • 4'b0010: OUT data packet received • 4'b0011: OUT transfer completed (triggers an interrupt) • 4'b0100: SETUP transaction completed (triggers an interrupt) • 4'b0110: SETUP data packet received • Others: Reserved Reset: 4'h0 Values: • 0x0 (DATA0): DATA0 • 0x3 (MDATA): MDATA • 0x2 (DATA1): DATA1 • 0x1 (DATA2): DATA2 Value After Reset: 0x0

16 15	DPID	Data PID (DPID) Indicates the Data PID of the received OUT data packet • 2'b00: DATA0 • 2'b10: DATA1 • 2'b01: DATA2 • 2'b11: MDATA Reset: 2'h0 Values: • 0x0 (DATA0): DATA0 • 0x3 (MDATA): MDATA • 0x2 (DATA1): DATA1 • 0x1 (DATA2): DATA2 Value After Reset: 0x0
14 4	BCnt	Byte Count (BCnt) Indicates the byte count of the received data packet. Value After Reset: 0x0
3 0	PNum	Endpoint Number (EPNum) Indicates the endpoint number to which the current received packet belongs. Values: • 0x2 (CHEP2): Channel or EndPoint 2 • 0x0 (CHEP0): Channel or EndPoint 0 • 0x3 (CHEP3): Channel or EndPoint 3 • 0x4 (CHEP4): Channel or EndPoint 4 • 0x1 (CHEP1): Channel or EndPoint 1 Value After Reset: 0x0

22.2.10 GRXFSIZ: Receive FIFO size register

The application can program the RAM size that must be allocated to the RxFIFO.

GRXSIZ =0x5000_0024

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			RxFDep
-			0x200
-			RW

10 0	RxFDep	RxFIFO Depth (RxFDep) This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth during configuration. You can write a new value in this field. Programmed values must not exceed the power-on value.

22.2.11 GNPTXFSIZ: Non-periodic transmit FIFO size register

The application can program the RAM size and the memory start address for the Non-periodic TxFIFO.

GNPTXFSIZ =0x5000_0028

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
INEPTxF0Dep		INEPTxF0StAddr	
0x0040		0x0200	
RW		RW	

26	INEPTxF0Dep	IN Endpoint TxFIFO 0 Depth (INEPTxF0Dep) This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 32,768 The application can write a new value in this field. Programmed values must not exceed the power-on value set in coreConsultant. The power-on reset value of this field is specified during coreConsultant configuration as Largest IN Endpoint FIFO 0 Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_0).
19	INEPTxF0StAddr	IN Endpoint FIFO0 Transmit RAM Start Address(INEPTxF0StAddr) This field contains the memory start address for IN Endpoint Transmit FIFO# 0. Programmed values must not exceed the power-on value.

22.2.12 GHWCFG1: User HW config1 register

GHWCFG1=0x5000_0032

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
EpDir			
0			
R			

31 0	EpDir	This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint • Bits[1:0]: Endpoint 0 direction (always BIDIR) • Bits [3:2]: Endpoint 1 direction, and so on. Direction • 2'b00: BIDIR (IN and OUT) endpoint • 2'b01: IN endpoint • 2'b10: OUT endpoint • 2'b11: Reserved

22.2.13 GHWCFG2: User HW config2 register

GHWCFG2=0x5000_0048

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	DynFifoSizing	Reserved	NumDevEps	FSPhyType	Reserved
-	1	-	0x4	1	-
-	R	-	R	R	-

19	DynFifoSizing	Dynamic FIFO Sizing Enabled (DynFifoSizing) • 1'b0: No • 1'b1: Yes Values: • 0x0 (DISABLED): Dynamic FIFO Sizing Disabled • 0x1 (ENABLED): Dynamic FIFO Sizing Enabled
13 10	NumDevEps	Number of Device Endpoints (NumDevEps) Indicates the number of device endpoints supported by the core. The range of this field is 0-4. Values: • 0x1 (ENDPT1): End point 1 • 0x4 (ENDPT4): End point 4 • 0x0 (ENDPT0): End point 0 • 0x2 (ENDPT2): End point 2 • 0x3 (ENDPT3): End point 3
9 8	FSPhyType	Full-Speed PHY Interface Type (FSPhyType) • 2'b01: Dedicated full-speed interface Values: • 0x1 (FS): Dedicated full-speed interface is supported Value After Reset: 2'b01

22.2.14 GHWCFG3: User HW config2 register

GHWCFG3=0x5000_004C

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
DfifoDepth		LPMMode	BCSupport	Reserved	RstType	Reserved	PktSizeWidth	XferSizeWidth
0x01D8		0	1	-	0	-	0x6	0x8
R		R	R	-	R	-	R	R

31 16	DfifoDepth	DFIFO Depth (DfifoDepth – EP_LOC_CNT) This value is in terms of 32-bit words. • Minimum value is 32 • Maximum value is 32,768
15	LPMMode	LPM mode specified for Mode of Operation. Values: • 0x0 (DISABLED): LPM disabled • 0x1 (ENABLED): LPM enabled

14	BCSupport	This bit indicates the controller support for Battery Charger. • 0 – No Battery Charger Support • 1 – Battery Charger support present Values: • 0x0 (DISABLED): No Battery Charger Support • 0x1 (ENABLED): Battery Charger Support present
11	RstType	Reset Style for Clocked always Blocks in RTL (RstType) • 1'b0: Asynchronous reset is used in the controller • 1'b1: Synchronous reset is used in the controller Values: • 0x0 (ASYNCRST): Asynchronous reset is used in the core • 0x1 (SYNCRST): Synchronous reset is used in the core
6 4	PktSizeWidth	Width of Packet Size Counters (PktSizeWidth) • 3'b000: 4 bits • 3'b001: 5 bits • 3'b010: 6 bits • 3'b011: 7 bits • 3'b100: 8 bits • 3'b101: 9 bits • 3'b110: 10 bits • Others: Reserved Values: • 0x2 (BITS6): Width of Packet Size Counter 6 • 0x1 (BITS5): Width of Packet Size Counter 5 • 0x6 (BITS10): Width of Packet Size Counter 10 • 0x5 (BITS9): Width of Packet Size Counter 9 • 0x3 (BITS7): Width of Packet Size Counter 7 • 0x4 (BITS8): Width of Packet Size Counter 8 • 0x0 (BITS4): Width of Packet Size Counter 4
3 0	XferSizeWidth	Width of Transfer Size Counters (XferSizeWidth) • 4'b0000: 11 bits • 4'b0001: 12 bits ... • 4'b1000: 19 bits • Others: Reserved Values: • 0x2 (WIDTH13): Width of Transfer Size Counter 13 bits • 0x3 (WIDTH14): Width of Transfer Size Counter 14 bits • 0x1 (WIDTH12): Width of Transfer Size Counter 12 bits • 0x8 (WIDTH19): Width of Transfer Size Counter 19 bits • 0x0 (WIDTH11): Width of Transfer Size Counter 11 bits • 0x7 (WIDTH18): Width of Transfer Size Counter 18 bits • 0x6 (WIDTH17): Width of Transfer Size Counter 17 bits • 0x5 (WIDTH16): Width of Transfer Size Counter 16 bits • 0x4 (WIDTH15): Width of Transfer Size Counter 15 bits

22.2.15 GHWCFG4: User hardware config4 register

GHWCFG4=0x5000_0050

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
DescDMA	DescDMAEnable	INEps	Reserved	NumCtlEps	Reserved	Hibernation	AhbFreq	PartialPwrDn	NumDevPerioEps
1	1	0x4	-	0	-	0	1	1	0
R	R	R	-	R	-	R	R	R	R

31	DescDMA	Scatter/Gather DMA configuration • 1'b0: Non Dynamic configuration • 1'b1: Dynamic configuration Values: • 0x0 (CONFIG1): Non Dynamic configuration • 0x1 (CONFIG2): Dynamic configuration
30	DescDMAEnabled	Scatter/Gather DMA configuration • 1'b0: Non-Scatter/Gather DMA configuration • 1'b1: Scatter/Gather DMA configuration Values: • 0x0 (DISABLE): Non-Scatter/Gather DMA configuration • 0x1 (ENABLE): Scatter/Gather DMA configuration
29 26	INEps	Number of Device Mode IN Endpoints Including Control Endpoints (INEps) • 0: 1 IN Endpoint • 1: 2 IN Endpoints .... • 4: 5 IN Endpoints Values: • 0x0 (ENDPT1): In Endpoint 1 • 0x1 (ENDPT2): In Endpoint 2 • 0x2 (ENDPT3): In Endpoint 3 • 0x3 (ENDPT4): In Endpoint 4
19 16	NumCtlEps	Number of Device Mode Control Endpoints in Addition to Endpoint 0 (NumCtlEps) Range: 0-4 Values: • 0x1 (ENDPT1): End point 1 • 0x4 (ENDPT4): End point 4 • 0x0 (ENDPT0): End point 0 • 0x2 (ENDPT2): End point 2 • 0x3 (ENDPT3): End point 3
6	Hibernation	Enable Hibernation (Hibernation) • 1'b0: Hibernation feature not enabled • 1'b1: Hibernation feature enabled Values: • 0x0 (DISABLED): Hibernation feature disabled • 0x1 (ENABLED): Hibernation feature enabled

5	AhbFreq	Minimum AHB Frequency Less Than 60 MHz (AhbFreq) • 1'b0: No • 1'b1: Yes Values: • 0x0 (DISABLED): Minimum AHB Frequency More Than 60 MHz • 0x1 (ENABLED): Minimum AHB Frequency Less Than 60 MHz
4	PartialPwrDn	Enable Partial Power Down (PartialPwrDn) • 1'b0: Partial Power Down Not Enabled • 1'b1: Partial Power Down Enabled Values: • 0x0 (DISABLED): Partial Power Down disabled • 0x1 (ENABLED): Partial Power Down enabled
3 0	NumDevPerioEps	Number of Device Mode Periodic IN Endpoints (NumDevPerioEps) Range: 0-15

22.2.16 GLPMCFG: LPM config register

GLPMCFG =0x5000_0054

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	LPM_RestoreSlpSts	LPM_EnBESL	LPM_RetryCnt_Sts	Reserved	LPM_Retry_Cnt	Reserved	L1ResumeOK	SlpSts	CoreL1Res	HIRD_Thres	EnblSlpM	bRemoteWake	HIRD	AppL1Res	LPMCap
-	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
-	RW	RW	R	-	RW	-	R	R	R	RW	RW	R	R	RW	RW

29	LPM_RestoreSlpSts	LPM Restore Sleep Status (LPM_RestoreSlpSts) When the application power gates the core (Partial Power Down / Hibernation) the application needs to program this bit to restore the LPM status in the core. The application needs to program this bit, during restore process, based on whether it had decided to go into Shallow Sleep (Clock Gating Only) or Deep Sleep (Power Gating) based on the BESL value received from the Host • 1'b0: The application puts the core in Shallow Sleep based on BESL value from the Host • 1'b1: The application puts the core in Deep Sleep based on BESL value from the Host Values: • 0x0 (DISABLED): Puts the core in Shallow Sleep mode based on the BESL value from the Host • 0x1 (ENABLED): Puts the core in Deep Sleep mode based on the BESL value from the Host Value After Reset: 0x0
28	LPM_EnBESL	LPM Enable BESL (LPM_EnBESL) This bit enables the BESL feature as defined in LPM Errata • 1'b0: The core works as per USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification as of July 16, 2007 • 1'b1: The core works as per the LPM Errata Values: • 0x0 (DISABLED): BESL is disabled • 0x1 (ENABLED): BESL is enabled as defined in LPM Errata Value After Reset: 0x0

27 25	LPM_RetryCnt_Sts	LPM Retry Count Status (LPM_RetryCnt_Sts) Number of LPM Host Retries still remaining to be transmitted for the current LPM sequence. Values: • 0x3 (RETRY_REM3): Three LPM retries remaining • 0x4 (RETRY_REM4): Four LPM retries remaining • 0x2 (RETRY_REM2): Two LPM retries remaining • 0x1 (RETRY_REM1): One LPM retry remaining • 0x6 (RETRY_REM6): Six LPM retries remaining • 0x5 (RETRY_REM5): Five LPM retries remaining • 0x7 (RETRY_REM7): Seven LPM retries remaining • 0x0 (RETRY_REM0): Zero LPM retries remaining Value After Reset: 0x0
23 21	LPM_Retry_Cnt	LPM Retry Count (LPM_Retry_Cnt) Number of additional LPM retries that the HOST would perform if the Device Response was an ERROR until a valid device response is received (STALL/NYET/ACK). Values: • 0x3 (RETRY3): Three LPM retries • 0x1 (RETRY1): One LPM retry • 0x2 (RETRY2): Two LPM retries • 0x0 (RETRY0): Zero LPM retries • 0x7 (RETRY7): Seven LPM retries • 0x4 (RETRY4): Four LPM retries • 0x6 (RETRY6): Six LPM retries • 0x5 (RETRY5): Five LPM retries Value After Reset: 0x0
16	L1ResumeOK	Sleep State Resume OK (L1ResumeOK) Indicates that the application or host can start resume from Sleep state. This bit is valid in LPM sleep (L1) state. It is set in sleep mode after a delay of 50 micro sec (TL1Residency). The bit is reset when SlpSts is 0. • 1'b0: The application/core cannot start resume from Sleep state. • 1'b1: The application/core can start resume from Sleep state. Values: • 0x0 (NOTOK): The application/core cannot start Resume from Sleep state • 0x1 (OK): The application/core can start Resume from Sleep state Value After Reset: 0x0
15	SlpSts	Port Sleep Status (SlpSts) This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the expiry of timer TL1TokenRetry. The core comes out of sleep: • When there is any activity on the USB line_state • When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device. Values: • 0x1 (CORE_IN_L1): In Device mode, the core enters the Sleep state when an ACK response is sent to an LPM transaction • 0x0 (CORE_NOT_IN_L1): In Device mode, this bit indicates core is not in L1 Value After Reset: 0x0

14 13	CoreL1Res	LPM response (CoreL1Res) The response of the core to LPM transaction received is reflected in these two bits. Values: • 0x0 (LPMRESP1): ERROR : No handshake response • 0x1 (LPMRESP2): STALL response • 0x2 (LPMRESP3): NYET response • 0x3 (LPMRESP4): ACK response Value After Reset: 0x0
12 8	HIRD_Thres	BESL/HIRD Threshold (HIRD_Thres) • EnBESL = 1'b0: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD value is greater than or equal to the value defined in this field HIRD_Thres[3:0] and HIRD_Thres[4] is set to 1b1. • EnBESL = 1'b1: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when BESL value is greater than or equal to the value defined in this field BESL_Thres[3:0] and BESL_Thres [4] is set to 1'b1. Value After Reset: 0x0
7	EnblSlpM	Enable utmi_sleep_n (EnblSlpM) The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. • 1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. • 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted. Values: • 0x0 (DISABLED): utmi_sleep_n assertion from the core is not transferred to the external PHY • 0x1 (ENABLED): utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted Value After Reset: 0x0
6	bRemoteWake	RemoteWakeEnable (bRemoteWake) This field is read only. It is updated with the Received LPM Token bRemoteWake bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. Values: • 0x0 (DISABLED): Remote Wakeup is disabled • 0x1 (ENABLED): In device mode, this field takes the value of remote wake up Value After Reset: 0x0
5 2	HIRD	Host-Initiated Resume Duration (HIRD) • EnBESL = 1'b0 Host Initiated Resume Duration. This field is read only and is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. • EnBESL = 1'b1 Best Effort Service Latency (BESL). This field is updated with the Received LPM Token BESL bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. Value After Reset: 0x0

1	AppL1Res	LPM response programmed by application (AppL1Res) Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core always responds with a NYET. If GLPMCFG.LPMCap is 1'b1, the core responds as follows: • 1: ACK Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: -There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR) -A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL) -No data is pending in the Transmit queue (else NYET) • 0: NYET The pre-programmed software bit is overridden for response to LPM token when: -The received bLinkState is not L1 (STALL response) -An error is detected in either of the LPM token packets due to corruption (ERROR response). Values: • 0x1 (ACK_RESP): The core responds with an ACK only on a successful LPM transaction • 0x0 (NYET_RESP): The core responds with a NYET when an error is detected in either of the LPM token packets due to corruption Value After Reset: 0x0
0	LPMCap	LPM-Capable (LPMCap) The application uses this bit to control the controller LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device/hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. • 1'b0: LPM capability is not enabled. • 1'b1: LPM capability is enabled. Values: • 0x0 (DISABLED): LPM capability is not enabled • 0x1 (ENABLED): LPM capability is enabled Value After Reset: 0x0

22.2.17 GPWRDN: Global power down register

GPWRDN =0x5000_0058

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	MultValIdBC	Reserved	BsessVld	Reserved	LineState	StsChngIntMsk	StsChngInt	Reserved	ConnDetMsk	ConnectDet	DisconnectDetect	DisconnectDetect	ResetDetMsk	ResetDetected	LineStageChange	LnStsChng	Reserved	PwrDnSwtch	PwrDnRst_n	PwrDnClmp	Restore	PMUActv	PMUIntSel
-	0	-	0	-	0	0	0	-	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0
-	R	-	R	-	R	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW

28 24	MultValIdBC	MultValIdBC Battery Charger ACA inputs in the following order: • Bit 26 – rid_float • Bit 25 – rid_gnd • Bit 24 – rid_a • Bit 23 – rid_b • Bit 22 – rid_c These bits are present only if BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0. Values: • 0xc (RID_A_RID_GND): OTG device as A-device, RID_A=1 and RID_GND=1 • 0x4 (RID_A): OTG device as A-device • 0x1f (RID_1): OTG device as A-device • 0x10 (RID_FLOAT): ID_OTG pin is floating • 0x0 (RID_0): OTG device as B-device • 0x8 (RID_GND): ID_OTG pin is grounded • 0x2 (RID_B): OTG device as B-device, cannot connect • 0x12 (RID_B_RID_FLOAT): OTG device as B-device, cannot connect, RID_B=1 and RID_FLOAT=1 • 0x11 (RID_C_RID_FLOAT): OTG device as B-device, can connect, RID_C=1 and RID_FLOAT=1 • 0x1 (RID_C): OTG device as B-device, can connect Value After Reset: 0x0
22	BsessVld	B Session Valid (BsessVld) This field reflects the B session valid status signal from the PHY. • 1'b0: B-Valid is 0. • 1'b1: B-Valid is 1. This bit is valid only when GPWRDN.PMUActv is 1. Values: • 0x0 (NOTVALID): B_Valid is 0 • 0x1 (VALID): B_Valid is 1 Value After Reset: 0x0
20 19	LineState	LineState This field indicates the current linestate on USB as seen by the PMU module. • 2'b00: DM = 0, DP = 0. • 2'b01: DM = 0, DP = 1. • 2'b10: DM = 1, DP = 0. • 2'b11: Not-defined. This bit is valid only when GPWRDN.PMUActv is 1. Values: • 0x0 (LS1): Linestate on USB: DM = 0, DP = 0 • 0x2 (LS3): Linestate on USB: DM = 1, DP = 0 • 0x1 (LS2): Linestate on USB: DM = 0, DP = 1 • 0x3 (LS4): Linestate on USB: Not-defined Value After Reset: 0x0

18	StsChngIntMsk	StsChngIntMsk Mask for StsChng Interrupt Values: • 0x1 (MASK): Mask for Status Change Interrupt • 0x0 (NOMASK): No Status Change Interrupt Mask Value After Reset: 0x0
17	StsChngInt	Status Change Interrupt (StsChngInt) This field indicates a status change in either the IDDIG or BsessVld signal. • 1'b0: No Status change • 1'b1: Status change detected After receiving this interrupt the application should read the GPWRDN register and interpret the change in BsesVld with respect to the previous value stored by the application. Values: • 0x0 (DISABLED): No Status change • 0x1 (ENABLED): Status change detected Value After Reset: 0x0
14	ConnDetMsk	ConnDetMsk Mask for ConnectDet interrupt Values: • 0x1 (MASK): Mask for ConnectDet Interrupt • 0x0 (NOMASK): No ConnectDet Interrupt Mask Value After Reset: 0x0
13	ConnectDet	ConnectDet This field indicates that a new connect has been detected • 1'b0: Connect not detected • 1'b1: Connect detected Values: • 0x0 (DISABLED): Connect not detected • 0x1 (ENABLED): Connect detected Value After Reset: 0x0
12	DisconnectDetect Msk	DisconnectDetectMsk Mask For DisconnectDetect Interrupt Values: • 0x1 (MASK): Mask for DisconnectDetect Interrupt • 0x0 (NOMASK): No DisconnectDetect Interrupt Mask Value After Reset: 0x0
11	DisconnectDetect	DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. • 1'b0: Disconnect not detected • 1'b1: Disconnect detected Values: • 0x0 (DISABLED): Disconnect not detected • 0x1 (ENABLED): Disconnect detected Value After Reset: 0x0
10	ResetDetMsk	ResetDetMsk Mask for ResetDetected interrupt Values: • 0x1 (MASK): Mask for ResetDetect Interrupt • 0x0 (NOMASK): No ResetDetect Interrupt Mask Value After Reset: 0x0

9	ResetDetected	ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. • 1'b0: Reset Not Detected • 1'b1: Reset Detected Values: • 0x0 (DISABLED): Reset not detected • 0x1 (ENABLED): Reset detected Value After Reset: 0x0
8	LineStageChange Msk	LineStageChangeMsk Mask for LineStateChange interrupt. Values: • 0x1 (MASK): Mask for LineStateChange Interrupt • 0x0 (NOMASK): No LineStateChange Interrupt Mask Value After Reset: 0x0
7	LnStsChng	Line State Change (LnStsChng) This interrupt is asserted when there is a Linestate Change detected by the PMU. The application should read GPWRDN.Linestate to determine the current linestate on USB. • 1'b0: No LineState change on USB • 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1. Values: • 0x0 (DISABLED): No LineState change on USB • 0x1 (ENABLED): LineState change on USB Value After Reset: 0x0
5	PwrDnSwtch	Power Down Switch (PwrDnSwtch) This bit indicates to the controller whether the VDD switch is in ON/OFF state. • 1'b0: The controller is in ON state • 1'b1: The controller is in OFF state Note: This bit must not be written to during normal mode of operation. Values: • 0x1 (OFF): The controller is in OFF state • 0x0 (ON): The controller is in ON state Value After Reset: 0x0
4	PwrDnRst_n	Power Down ResetN (PwrDnRst_n) The application must program this bit to reset the core during the Hibernation exit process. • 1'b1: The controller is in normal operation • 1'b0: reset the controller Note: This bit must not be written to during normal mode of operation. Values: • 0x0 (DISABLE): Reset the controller • 0x1 (ENABLE): The controller is in normal operation Value After Reset: 0x1
3	PwrDnClmp	Power Down Clamp (PwrDnClmp) The application must program this bit to enable or disable the clamps to all the outputs of the core module to prevent the corruption of other active logic. • 1'b0: Disable PMU power clamp • 1'b1: Enable PMU power clamp Values: • 0x0 (DISABLE): Disable PMU power clamp • 0x1 (ENABLE): Enable PMU power clamp Value After Reset: 0x0

2	Restore	Restore The application should program this bit to enable or disable restore mode from the PMU module. • 1'b0: The controller in normal mode of operation • 1'b1: The controller in restore mode Note: This bit must not be written to during normal mode of operation. Values: • 0x0 (DISABLE): The controller in normal mode of operation • 0x1 (ENABLE): The controller in Restore mode Value After Reset: 0x0
1	PMUActv	PMU Active (PMUActv) This is bit is to enable or disable the PMU logic. • 1'b0: Disable PMU module • 1'b1: Enable PMU module Note: This bit must not be written to during normal mode of operation. Values: • 0x0 (DISABLE): Disable PMU module • 0x1 (ENABLE): Enable PMU module Value After Reset: 0x0
0	PMUIntSel	PMU Interrupt Select (PMUIntSel) A write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state all interrupts from the DWC_usb11_device_core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation. • 1'b0: Internal DWC_usb11_device_core interrupt is selected • 1'b1: External DWC_usb11_device_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation. Values: • 0x0 (DISABLE): Internal DWC_usb11_device_core interrupt is selected • 0x1 (ENABLE): External DWC_usb11_device_pmu interrupt is selected Value After Reset: 0x0

22.2.18 GDFIFOCFG: Global DFIFO configuration register

GDFIFOCFG =0x5000_005C

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
EPInfoBaseAdd		GDFIFOCfg	
0x01D8		0x0200	
RW		RW	

31 16	EPInfoBaseAddr	EPInfoBaseAddr This field provides the start address of the EP info controller
15 0	GDFIFOCfg	GDFIFOCfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non-zero value to this register. The value programmed must conform to the guidelines described in 'FIFO RAM Allocation'. The core does not have any corrective logic if the FIFO sizes are programmed incorrectly.

22.2.19 DIEPTXFi: Device IN endpoint transmit FIFO size register i

For I = 1; I <= OTG_NUM_IN_EPS

DIEPTXFi =0x5000_013C

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	INEPnTxFDep	INEPnTxFStAddr	
-	0	0	
-	RW	RW	

27 16	INEPnTxFDep	IN Endpoint TxFIFO Depth (INEPnTxFDep) This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value
15 0	INEPnTxFStAddr	IN Endpoint FIFOn Transmit RAM Start Address (INEPnTxFStAddr) This field contains the memory start address for IN endpoint Transmit FIFOn (0<n< = 15). The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. Programmed values must not exceed the power-on value.

22.2.20 DCFG: Device configuration register

DCFG =0x5000_0800

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
ResValid	PerSchIntvl	DescDMA	Reserved	ErraticIntMsk	Reserved	EnDevOutNa	PerFrInt	DevAddr	Ena32KhzSu	NZStsOUTHS	DevSpd
0x02	0	0	-	0	-	0	0	0	0	0	0
RW	RW	RW	-	RW	-	RW	RW	RW	RW	RW	RW

31 26	ResValid	Resume Validation Period (ResValid) This field is effective only when DCFG.Ena32KhzSusp is set. It will control the resume period when the core resumes from suspend. The core counts for ResValid number of clock cycles to detect a valid resume when this is set Value After Reset: 0x2
25 24	PerSchIntvl	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25, 50 or 75% of (micro) frame. • When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. • When no periodic endpoints are active, Then the internal DMA engine services non-periodic endpoints, ignoring this field. • After the specified time within a (micro) frame, the DMA switches to fetching for non-periodic endpoints. -2'b00: 25% of (micro) frame. -2'b01: 50% of (micro) frame. -2'b10: 75% of (micro) frame. -2'b11: Reserved. Reset: 2'b00 Values: • 0x0 (MF25): 25% of (micro)frame • 0x2 (MF75): 75% of (micro)frame • 0x3 (RESERVED): Reserved • 0x1 (MF50): 50% of (micro)frame Value After Reset: 0x0
23	DescDMA	Enable Scatter/gather DMA in device mode (DescDMA). When the Scatter/Gather DMA option selected during configuration of the RTL, the application can Set this bit during initialization to enable the Scatter/Gather DMA operation. Note: This bit must be modified only once after a reset. The following combinations are available for programming: • GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Invalid • GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid • GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Invalid • GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode Values: • 0x0 (DISABLED): Disable Scatter/Gather DMA • 0x1 (ENABLED): Enable Scatter/Gather DMA Value After Reset: 0x0

15	ErraticIntMsk	Erratic Error Interrupt Mask • 1'b1: Mask early suspend interrupt on erratic error • 1'b0: Early suspend interrupt is generated on erratic error Values: • 0x1 (MASK): Mask early suspend interrupt on erratic error • 0x0 (NOMASK): Early suspend interrupt is generated on erratic error Value After Reset: 0x0
13	EnDevOutNak	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA • 1'b0 : The core does not set NAK after Bulk OUT transfer complete • 1'b1 : The core sets NAK after Bulk OUT transfer complete It is one time programmable after reset like any other DCFG register bits. Values: • 0x0 (DISABLED): The core does not set NAK after Bulk OUT transfer complete • 0x1 (ENABLED): The core sets NAK after Bulk OUT transfer complete Value After Reset: 0x0
12 11	PerFrInt	Periodic Frame Interval (PerFrInt) Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine If all the isochronous traffic for that (micro) frame is complete. • 2'b00: 80% of the (micro)frame interval • 2'b01: 85% • 2'b10: 90% • 2'b11: 95% Values: • 0x0 (EOPF80): 80% of the (micro)frame interval • 0x1 (EOPF85): 85% of the (micro)frame interval • 0x3 (EOPF95): 95% of the (micro)frame interval • 0x2 (EOPF90): 90% of the (micro)frame interval Value After Reset: 0x0
10 4	DevAddr	Device Address (DevAddr) The application must program this field after every SetAddress control command. Value After Reset: 0x0
3	Ena32KhzSusp	Enable 32 KHz Suspend mode (Ena32KhzSusp) As FS PHY interface is chosen, when this bit is set, the core expects that the PHY clock during Suspend is switched from 48 MHz to 32 KHz. Values: • 0x0 (DISABLED): USB 1.1 Full-Speed Serial Transceiver not selected • 0x1 (ENABLED): USB 1.1 Full-Speed Serial Transceiver Interface selected Value After Reset: 0x0

2	NZStsOUTHShk	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. • 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. • 1'b0: Send the received OUT packet to the application (zero-length or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. Values: • 0x0 (SENDOUT): Send the received OUT packet to the application (zero-length or non-zero length) and send a handshake based on NAK and STALL bits for the endpoint in the Device Endpoint Control Register • 0x1 (SENDSTALL): Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application Value After Reset: 0x0
1 0	DevSpd	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the connect sequence is completed, and is based on the speed of the USB host to which the core is connected. Values: • 0x0 (Reserved0): Reserved • 0x1 (Reserved1): Reserved • 0x3 (USBFS1148): Full speed USB 1.1 transceiver clock is 48 MHz • 0x2 (USBLS116): Low speed USB 1.1 transceiver clock is 6 MHz Value After Reset: 0x0

22.2.21 DCTL: Device control register

DCTL =0x5000_0804

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		DeepSleepBESLR	EnContOnBNA	NakOnBble	IgnrFrmNum	GMC		Reserved	PWROnPrgDone	CGOUTNak	SGOUTNak	CGNPInNak	SGNPInNak	TstCtl			GOUTNakSts	GNPINNakSts	SftDiscon	RmtWkUpSig
-		0	0	0	0	0		-	0	0	0	0	0	0			0	0	1	0
-		RW	RW	RW	RW	RW		-	W	W	W	W	RW	RW			R	R	RW	RW

18	DeepSleepBESLReject	DeepSleepBESLReject • 1: Deep Sleep BESL Reject feature is enabled • 0: Deep Sleep BESL Reject feature is disabled When enabled Core rejects LPM request with HIRD value greater than HIRD threshold programmed. NYET response is sent for LPM tokens with HIRD value greater than HIRD threshold. By default, the Deep Sleep BESL Reject feature is disabled. Values: • 0x0 (DISABLED): Deep Sleep BESL Reject feature is disabled • 0x1 (ENABLED): Deep Sleep BESL Reject feature is enabled Value After Reset: 0x0
17	EnContOnBNA	Enable Continue on BNA (EnContOnBNA) This bit enables the core to continue on BNA for Bulk OUT endpoints. With this feature enabled, when a Bulk OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. • 1'b0: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor. • 1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt. It is a one-time programmable after reset bit like any other DCTL register bits. Values: • 0x0 (DISABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the DOEPDMA descriptor • 0x1 (ENABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the descriptor that received the BNA interrupt. Value After Reset: 0x0
16	NakOnBble	NAK on Babble Error (NakOnBble) Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received. Values: • 0x0 (DISABLED): Disable NAK on Babble Error • 0x1 (ENABLED): NAK on Babble Error Value After Reset: 0x0

15	IgnrFrmNum	Ignore Frame number for Isochronous End points (IgnrFrmNum) Do not program IgnrFrmNum bit to 1'b1 when the core is operating in threshold mode. When this bit is enabled, there must be only one packet per descriptor. • 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. • 1: The core ignores the frame number, sending packets immediately as the packets are ready. In Scatter/Gather DMA mode, if this bit is enabled, the packets are not flushed when an ISOC IN token is received for an elapsed frame. Values: • 0x0 (DISABLED): The core transmits the packets only in the frame number in which they are intended to be transmitted • 0x1 (ENABLED): The core ignores the frame number, sending packets immediately as the packets are ready Value After Reset: 0x0
14 13	GMC	Global Multi Count (GMC) GMC must be programmed only once after initialization. This field indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic endpoints. • 2'b00: Invalid. • 2'b01: 1 packet. • 2'b10: 2 packets. • 2'b11: 3 packets. Values: • 0x0 (NOTVALID): Invalid • 0x1 (ONEPACKET): 1 packet • 0x2 (TWOPACKET): 2 packets • 0x3 (THREEPACKET): 3 packets Value After Reset: 0x0
11	PWROnPrgDone	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode. Values: • 0x1 (DONE): Power-On Programming Done • 0x0 (NOTDONE): Power-On Programming not done Value After Reset: 0x0
10	CGOUTNak	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK. Values: • 0x0 (DISABLED): Disable Clear Global OUT NAK • 0x1 (ENABLED): Clear Global OUT NAK Value After Reset: 0x0
9	SGOUTNak	Set Global OUT NAK (SGOUTNak) A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared. Values: • 0x0 (DISABLED): Disable Global OUT NAK • 0x1 (ENABLED): Set Global OUT NAK Value After Reset: 0x0

8	CGNPInNak	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK. Values: • 0x0 (DISABLE): Disable Global Non-periodic IN NAK • 0x1 (ENABLE): Clear Global Non-periodic IN NAK Value After Reset: 0x0
7	SGNPInNak	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also Set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must Set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared Values: • 0x0 (DISABLE): Disable Global Non-periodic IN NAK • 0x1 (ENABLE): Set Global Non-periodic IN NAK Value After Reset: 0x0
6 4	TstCtl	Test Control (TstCtl) • 3'b000: Test mode disabled • 3'b001: Test_J mode • 3'b010: Test_K mode • 3'b011: Test_SE0_NAK mode • 3'b100: Test_Packet mode • 3'b101: Test_Force_Enable • Others: Reserved Values: • 0x2 (TESTK): Test_K mode • 0x4 (TESTPM): Test_Packet mode • 0x0 (DISABLED): Test mode disabled • 0x5 (TESTFE): Test_force_Enable • 0x1 (TESTJ): Test_J mode • 0x3 (TESTSN): Test_SE0_NAK mode Value After Reset: 0x0
3	GOUTNakSts	Global OUT NAK Status (GOUTNakSts) • 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. • 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped. Values: • 0x1 (ACTIVE): No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped. • 0x0 (INACTIVE): A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. Value After Reset: 0x0
2	GNPINNakSts	Global Non-periodic IN NAK Status (GNPINNakSts) • 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. • 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO. Values: • 0x1 (ACTIVE): A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO. • 0x0 (INACTIVE): A handshake is sent out based on the data availability in the transmit FIFO Value After Reset: 0x0

1	SftDiscon	Soft Disconnect (SftDiscon) The application uses this bit to signal the controller to do a soft disconnect. As long as this bit is Set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. • 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. • 1'b1: The core generates a device disconnect event to the USB host. Note: • This bit is not impacted by a soft reset. Values: • 0x1 (DISCONNECT): The core generates a device disconnect event to the USB host • 0x0 (NODISCONNECT): The core generates a device connect event to the USB host Value After Reset: 0x1
0	RmtWkUpSig	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must Set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it. Remote Wakeup Signaling (RmtWkUpSig) When LPM is enabled, In L1 state the behavior of this bit is as follows: When the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware will automatically clear this bit after a time of 50 micro sec (TL1DevDrvResume) after set by application. Application should not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction was zero. Values: • 0x0 (DISABLEDRMWKUP): Core does not send Remote Wakeup Signaling • 0x1 (ENABLERMWKUP): Core sends Remote Wakeup Signaling Value After Reset: 0x0

22.2.22 DSTS: Device status register

DSTS =0x5000_0808

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	DevLnSts	SOFFN	Reserved	ErrticErr	EnumSpd		SuspSts
-	0	0	-	0	1		0
-	R	R	-	R	R		R

23 22	DevLnSts	Device Line Status (DevLnSts) Indicates the current logic level USB data lines • DevLnSts[1]: Logic level of D+ • DevLnSts[0]: Logic level of D- Value After Reset: 0x0
21 8	SOFFN	Number of the Received SOF (SOFFN) Note: This register may return a non-zero value if read immediately after power-on reset. In case the register bit reads non-zero immediately after power-on reset, it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established. Value After Reset: 0x0
3	ErrticErr	Erratic Error (ErrticErr) The core sets this bit to report any erratic errors due to PHY error). Because of erratic errors, the controller goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover. Values: • 0x1 (ACTIVE): Erratic Error • 0x0 (INACTIVE): No Erratic Error Value After Reset: 0x0
2 1	EnumSpd	Enumerated Speed (EnumSpd) Indicates the speed at which the controller has come up after speed detection through a chirp sequence. • 2'b10: Low speed (PHY clock is running at 6 MHz) • 2'b11: Full speed (PHY clock is running at 48 MHz) Values: • 0x3 (FS48): Full speed (PHY clock is running at 48 MHz) • 0x0 (Reserved0): Reserved • 0x2 (LS6): Low speed (PHY clock is running at 6 MHz) • 0x1 (Reserved1): Reserved Value After Reset: 0x1
0	SuspSts	Suspend Status (SuspSts) In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend • When there is any activity on the phy_line_state_i signal, or • When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig). When the core comes out of the suspend, this bit is set to 1'b0. Values: • 0x1 (ACTIVE): Suspend state • 0x0 (INACTIVE): No suspend state Value After Reset: 0x0

22.2.23 DIEPMSK: Device IN endpoint common interrupt mask register

DIEPMSK =0x5000_0810

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			NAKMsk	Reserved	BNAInIntrMsk	TxfifoUndrnMsk	Reserved	INEPNakEffMsk	INTknEPMisMsk	INTknTXFEmpM	TimeOUTMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk
-			0	-	0	0	-	0	0	0	0	0	0	0
-			RW	-	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

13	NAKMsk	NAK interrupt Mask (NAKMsk) Values: • 0x0 (MASK): Mask NAK Interrupt • 0x1 (NOMASK): No Mask NAK Interrupt Value After Reset: 0x0
9	BNAInIntrMsk	BNA interrupt Mask (BNAInIntrMsk) Values: • 0x0 (MASK): Mask BNA Interrupt • 0x1 (NOMASK): No BNA Interrupt Mask Value After Reset: 0x0
8	TxfifoUndrnMsk	Fifo Underrun Mask (TxfifoUndrnMsk) Values: • 0x0 (MASK): Mask Fifo Underrun Interrupt • 0x1 (NOMASK): No Fifo Underrun Interrupt Mask Value After Reset: 0x0
6	INEPNakEffMsk	IN Endpoint NAK Effective Mask (INEPNakEffMsk) Values: • 0x0 (MASK): Mask IN Endpoint NAK Effective Interrupt • 0x1 (NOMASK): No IN Endpoint NAK Effective Interrupt Mask Value After Reset: 0x0
5	INTknEPMisMsk	IN Token received with EP Mismatch Mask (INTknEPMisMsk) Values: • 0x0 (MASK): Mask IN Token received with EP Mismatch Interrupt • 0x1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt Value After Reset: 0x0
4	INTknTXFEmpMsk	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk) Values: • 0x0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt • 0x1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt Value After Reset: 0x0
3	TimeOUTMsk	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints) Values: • 0x0 (MASK): Mask Timeout Condition Interrupt • 0x1 (NOMASK): No Timeout Condition Interrupt Mask Value After Reset: 0x0

2	AHBErrMsk	AHB Error Mask (AHBErrMsk) Values: • 0x0 (MASK): Mask AHB Error Interrupt • 0x1 (NOMASK): No AHB Error Interrupt Mask Value After Reset: 0x0
1	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: • 0x0 (MASK): Mask Endpoint Disabled Interrupt • 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask Value After Reset: 0x0
0	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: • 0x0 (MASK): Mask Transfer Completed Interrupt • 0x1 (NOMASK): No Transfer Completed Interrupt Mask Value After Reset: 0x0

22.2.24 DOEPMSK: Device OUT endpoint common interrupt mask register

DOEPMSK =0x5000_0814

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			NYETMsk	NAKMsk	BbleErrMsk	Reserved		BnaOutIntrMsk	OutPktErrMsk	Reserved	Back2BackSE	StsPhseRcvd	OUTTknEPdis	SetUPMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk
-			0	0	0	-		0	0	-	0	0	0	0	0	0	0
-			RW	RW	RW	-		RW	RW	-	RW	RW	RW	RW	RW	RW	RW

14	NYETMsk	NYET interrupt Mask (NYETMsk) Values: • 0x0 (MASK): Mask NYET Interrupt • 0x1 (NOMASK): No NYET Interrupt Mask Value After Reset: 0x0
13	NAKMsk	NAK interrupt Mask (NAKMsk) Values: • 0x0 (MASK): Mask NAK Interrupt • 0x1 (NOMASK): No NAK Interrupt Mask Value After Reset: 0x0
12	BbleErrMsk	Babble Error interrupt Mask (BbleErrMsk) Values: • 0x0 (MASK): Mask Babble Error Interrupt • 0x1 (NOMASK): No Babble Error Interrupt Mask Value After Reset: 0x0
9	BnaOutIntrMsk	BNA interrupt Mask (BnaOutIntrMsk) Values: • 0x0 (MASK): Mask BNA Interrupt • 0x1 (NOMASK): No BNA Interrupt Mask Value After Reset: 0x0
8	OutPktErrMsk	OUT Packet Error Mask (OutPktErrMsk) Values: • 0x0 (MASK): Mask OUT Packet Error Interrupt • 0x1 (NOMASK): No OUT Packet Error Interrupt Mask Value After Reset: 0x0

6	Back2BackSETup	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only. Values: • 0x0 (MASK): Mask Back-to-Back SETUP Packets Received Interrupt • 0x1 (NOMASK): No Back-to-Back SETUP Packets Received Interrupt Mask Value After Reset: 0x0
5	StsPhseRcvdMsk	Status Phase Received Mask (StsPhseRcvdMsk) Applies to control OUT endpoints only. Values: • 0x0 (MASK): Status Phase Received Mask • 0x1 (NOMASK): No Status Phase Received Mask Value After Reset: 0x0
4	OUTTknEPdisMsk	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only. Values: • 0x0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt • 0x1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask Value After Reset: 0x0
3	SetUPMsk	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only. Values: • 0x0 (MASK): Mask SETUP Phase Done Interrupt • 0x1 (NOMASK): No SETUP Phase Done Interrupt Mask Value After Reset: 0x0
2	AHBErrMsk	AHB Error (AHBErrMsk) Values: • 0x0 (MASK): Mask AHB Error Interrupt • 0x1 (NOMASK): No AHB Error Interrupt Mask Value After Reset: 0x0
1	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: • 0x0 (MASK): Mask Endpoint Disabled Interrupt • 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask Value After Reset: 0x0
0	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: • 0x0 (MASK): Mask Transfer Completed Interrupt • 0x1 (NOMASK): No Transfer Completed Interrupt Mask Value After Reset: 0x0

22.2.25 DAINT: Device all endpoints interrupt register

DAINT =0x5000_0818

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		OutEPInt4	OutEPInt3	OutEPInt2	OutEPInt1	OutEPInt0	Reserved		InEpInt4	InEpInt3	InEpInt2	InEpInt1	InEpInt0
-		0	0	0	0	0	-		0	0	0	0	0
-		R	R	R	R	R	-		R	R	R	R	R

20	OutEPInt4	OUT Endpoint 4 Interrupt Bit Value After Reset: 0x0
19	OutEPInt3	OUT Endpoint 3 Interrupt Bit Value After Reset: 0x0
18	OutEPInt2	OUT Endpoint 2 Interrupt Bit Value After Reset: 0x0
17	OutEPInt1	OUT Endpoint 1 Interrupt Bit Value After Reset: 0x0
16	OutEPInt0	OUT Endpoint 0 Interrupt Bit Values: • 0x1 (ACTIVE): OUT Endpoint 0 Interrupt • 0x0 (INACTIVE): No Interrupt Value After Reset: 0x0
4	InEpInt4	IN Endpoint 4 Interrupt Bit Value After Reset: 0x0
3	InEpInt3	IN Endpoint 3 Interrupt Bit Value After Reset: 0x0
2	InEpInt2	IN Endpoint 2 Interrupt Bit Value After Reset: 0x0
1	InEpInt1	IN Endpoint 1 Interrupt Bit Value After Reset: 0x0
0	InEpInt0	IN Endpoint 0 Interrupt Bit Values: • 0x1 (ACTIVE): IN Endpoint 0 Interrupt • 0x0 (INACTIVE): No Interrupt Value After Reset: 0x0

22.2.26 DAINTMSK: Device all endpoints interrupt mask register

DAINTMSK =0x5000_081C

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		OutEPInt4	OutEPInt3	OutEPInt2	OutEPInt1	OutEPInt0	Reserved		InEpInt4	InEpInt3	InEpInt2	InEpInt1	InEpInt0
-		0	0	0	0	0	-		0	0	0	0	0
-		RW	RW	RW	RW	RW	-		RW	RW	RW	RW	RW

20	OutEPMsk4	OUT Endpoint 4 Interrupt mask Bit Value After Reset: 0x0
19	OutEPMsk3	OUT Endpoint 3 Interrupt mask Bit Value After Reset: 0x0
18	OutEPMsk2	OUT Endpoint 2 Interrupt mask Bit Value After Reset: 0x0
17	OutEPMsk1	OUT Endpoint 1 Interrupt mask Bit Value After Reset: 0x0
16	OutEPMsk0	OUT Endpoint 0 Interrupt mask Bit Values: • 0x0 (MASK): OUT Endpoint 0 Interrupt mask • 0x1 (NOMASK): No Interrupt mask Value After Reset: 0x0
4	InEpMsk4	IN Endpoint 4 Interrupt mask Bit Value After Reset: 0x0
3	InEpMsk3	IN Endpoint 3 Interrupt mask Bit Value After Reset: 0x0
2	InEpMsk2	IN Endpoint 2 Interrupt mask Bit Value After Reset: 0x0
1	InEpMsk1	IN Endpoint 1 Interrupt mask Bit Value After Reset: 0x0
0	InEpMsk0	IN Endpoint 0 Interrupt mask Bit Values: • 0x0 (MASK): IN Endpoint 0 Interrupt mask • 0x1 (NOMASK): No Interrupt mask Value After Reset: 0x0

22.2.27 DTHRCTL: Device threshold control register

DTHRCTL =0x5000_0830

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	ArbPrkEn	Reserved	RxThrLen	RxThrEn	Reserved	AHBThrRatio	TxThrLen	ISOThrEn	NonISOThrEn
-	1	-	0x008	0	-	0	0x008	0	0
-	RW	-	RW	RW	-	RW	RW	RW	RW

27	ArbPrkEn	Arbiter Parking Enable (ArbPrkEn) This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is Set to one, Then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By Default the parking is enabled. Values: • 0x0 (DISABLED): Disable DMA arbiter parking • 0x1 (ENABLED): Enable DMA arbiter parking for IN endpoints Value After Reset: 0x1
25 17	RxThrLen	Receive Threshold Length (RxThrLen) This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HbstLen). Value After Reset: 0x8
16	RxThrEn	Receive Threshold Enable (RxThrEn) When this bit is set, the core enables thresholding in the receive direction. Note: We recommends that you do not enable RxThrEn, because it may cause issues in the RxFIFO especially during error conditions such as RxError and Babble. Values: • 0x0 (DISABLED): Disable thresholding • 0x1 (ENABLED): Enable thresholding in the receive direction Value After Reset: 0x0

12 11	AHBThrRatio	AHB Threshold Ratio (AHBThrRatio) These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements. • 2'b00: AHB threshold = MAC threshold • 2'b01: AHB threshold = MAC threshold / 2 • 2'b10: AHB threshold = MAC threshold / 4 • 2'b11: AHB threshold = MAC threshold / 8 Values: • 0x1 (THRESONE): AHB threshold = MAC threshold /2 • 0x2 (THRESTWO): AHB threshold = MAC threshold /4 • 0x3 (THRESTHREE): AHB threshold = MAC threshold /8 • 0x0 (THRESZERO): AHB threshold = MAC threshold Value After Reset: 0x0
10 2	TxThrLen	Transmit Threshold Length (TxThrLen) This field specifies Transmit thresholding size in DWORDS. This also forms the MAC threshold and specifies the amount of data in bytes to be in the corresponding endpoint transmit FIFO, before the core can start transmit on the USB. The threshold length has to be at least eight DWORDS when the value of AHBThrRatio is 2'h00. In case the AHBThrRatio is non zero the application needs to ensure that the AHB Threshold value does not go below the recommended eight DWORD. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HbstLen). Value After Reset: 0x8
1	ISOThrEn	ISO IN Endpoints Threshold Enable. (ISOThrEn) When this bit is Set, the core enables thresholding for isochronous IN endpoints. Values: • 0x0 (DISABLED): No thresholding • 0x1 (ENABLED): Enables thresholding for isochronous IN endpoints Value After Reset: 0x0
0	NonISOThrEn	Non-ISO IN Endpoints Threshold Enable. (NonISOThrEn) When this bit is Set, the core enables thresholding for Non Isochronous IN endpoints. Values: • 0x0 (DISABLED): No thresholding • 0x1 (ENABLED): Enable thresholding for non-isochronous IN endpoints Value After Reset: 0x0

22.2.28 DIEPEMPMSK: Device IN endpoint FIFO empty interrupt mask register

DIEPEMPMSK =0x5000_0834

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		InEpTxfEmpMsk	
-		0	
-		RW	

0	InEpTxfEmpMsk	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn.TxFEmp interrupt, one bit per IN Endpoint: Bit 0 for IN EP 0, bit 4 for IN EP 4 Values: • 0x8 (EP3_MASK): Mask IN EP3 Tx FIFO Empty Interrupt • 0x10 (EP4_MASK): Mask IN EP4 Tx FIFO Empty Interrupt • 0x1 (EP0_MASK): Mask IN EP0 Tx FIFO Empty Interrupt • 0x2 (EP1_MASK): Mask IN EP1 Tx FIFO Empty Interrupt • 0x4 (EP2_MASK): Mask IN EP2 Tx FIFO Empty Interrupt Value After Reset: 0x0

22.2.29 DIEPCTL0: Device control IN endpoint 0 control register

DIEPCTL0=0x5000_0900

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
EPEna	EPDis	Reserved	SNAK	CNAK	TxFNum		Stall	Reserved	EPType	NAKSts	Reserved	USBActEP	Reserved	MPS
0	0	-	0	0	0		0	-	0	0	-	1	-	0
RW	RW	-	W	W	RW		RW	-	R	R	-	R	-	RW

31	EPEna	Endpoint Enable (EPEna) When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. Values: • 0x1 (ACTIVE): Enable Endpoint • 0x0 (INACTIVE): No action Value After Reset: 0x0

30	EPDis	Endpoint Disable (EPDis) The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must Set this bit only if Endpoint Enable is already set for this endpoint. Values: • 0x1 (ACTIVE): Disabled Endpoint • 0x0 (INACTIVE): No action Value After Reset: 0x0
27	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. Values: • 0x0 (NOSET): No action • 0x1 (SET): Set NAK Value After Reset: 0x0
26	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: • 0x1 (CLEAR): Clear NAK • 0x0 (NOCLEAR): No action Value After Reset: 0x0
25 22	TxFNum	TxFIFO Number (TxFNum) • For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint. Values: • 0x3 (TXFIFO3): Tx FIFO 3 • 0x4 (TXFIFO4): Tx FIFO 4 • 0x0 (TXFIFO0): Tx FIFO 0 • 0x1 (TXFIFO1): Tx FIFO 1 • 0x2 (TXFIFO2): Tx FIFO 2 Value After Reset: 0x0
21	Stall	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Nonperiodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Values: • 0x1 (ACTIVE): Stall Handshake • 0x0 (INACTIVE): No Stall Value After Reset: 0x0
19 18	EPType	Endpoint Type (EPType) Hardcoded to 00 for control. Values: • 0x0 (ACTIVE): Endpoint Control 0 Value After Reset: 0x0
17	NAKSts	NAK Status (NAKSts) Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status • 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even If there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint • 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status Value After Reset: 0x0

15	USBActEP	USB Active Endpoint (USBActEP) This bit is always SET to 1, indicating that control endpoint 0 is always active in all configurations and interfaces. Values: • 0x1 (ACTIVE0): Control endpoint is always active Value After Reset: 0x1
1 0	MPS	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. • 2'b00: 64 bytes • 2'b01: 32 bytes • 2'b10: 16 bytes • 2'b11: 8 bytes Values: • 0x2 (BYTES16): 16 bytes • 0x1 (BYTES32): 32 bytes • 0x0 (BYTES64): 64 bytes • 0x3 (BYTES8): 8 bytes Value After Reset: 0x0

22.2.30 DIEPINT0: Device IN endpoint 0 interrupt register

DIEPINT0=0x5000_0908

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			NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn	TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEm	Reserved	AHBErr	EPDisbld	XferCompl
-			0	0	0	-		0	0	1	0	0	0	-	0	0	0
-			RW	RW	RW	-		RW	RW	R	RW	RW	RW	-	RW	RW	RW

14	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: • 0x1 (ACTIVE): NYET Interrupt • 0x0 (INACTIVE): No interrupt Value After Reset: 0x0
13	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. <brIn case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. Values: • 0x1 (ACTIVE): NAK Interrupt • 0x0 (INACTIVE): No interrupt Value After Reset: 0x0
12	BbleErr	NAK Interrupt (BbleErr) The core generates this interrupt when babble is received for the endpoint. Values: • 0x1 (ACTIVE): BbleErr interrupt • 0x0 (INACTIVE): No interrupt Value After Reset: 0x0

9	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as DMA done. Values: • 0x1 (ACTIVE): BNA interrupt • 0x0 (INACTIVE): No BNA interrupt Value After Reset: 0x0
8	TxfifoUndrn	Fifo Underrun (TxfifoUndrn) Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO underrun condition in threshold mode for this endpoint. Values: • 0x1 (ACTIVE): Fifo Underrun interrupt • 0x0 (INACTIVE): No Fifo Underrun interrupt Value After Reset: 0x0
7	TxFEmp	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). Values: • 0x1 (ACTIVE): Transmit FIFO Empty interrupt • 0x0 (INACTIVE): No Transmit FIFO Empty interrupt Value After Reset: 0x1
6	INEPNakEff	IN Endpoint NAK Effective (INEPNakEff) Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. Values: • 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt • 0x0 (INACTIVE): No IN Endpoint NAK Effective interrupt Value After Reset: 0x0
5	INTknEPMis	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Values: • 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt • 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt Value After Reset: 0x0
4	INTknTXFEmp	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. Values: • 0x1 (ACTIVE): IN Token Received when TxFIFO Empty Interrupt • 0x0 (INACTIVE): No IN Token Received when TxFIFO Empty interrupt Value After Reset: 0x0

2	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. Values: • 0x1 (ACTIVE): AHB Error interrupt • 0x0 (INACTIVE): No AHB Error Interrupt Value After Reset: 0x0
1	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: • 0x1 (ACTIVE): Endpoint Disabled Interrupt • 0x0 (INACTIVE): No Endpoint Disabled Interrupt Value After Reset: 0x0
0	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled -For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. -For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. Values: • 0x1 (ACTIVE): Transfer Completed Interrupt • 0x0 (INACTIVE): No Transfer Complete Interrupt Value After Reset: 0x0

22.2.31 DIEPDMA0: Device IN endpoint 0 DMA address register

DIEPDMA0=0x5000_0914

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
DMAAddr			
0			
RW			

31 0	DMAAddr	DMAAddr This field indicates the base pointer for the descriptor list

22.2.32 DTXFSTS0: Device IN endpoint transmit FIFO status register 0

DTXFSTS0=0x5000_0918

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		INEPTxFSpcAvail	
-		0x0040	
-		R	

15 0	INEPTxFSpcAvail	IN Endpoint TxFIFO Space Avail (INEPTxFSpcAvail) Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. • 16'h0: Endpoint TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16'hn: n words available (where 0 n 32,768) • 16'h8000: 32,768 words available • Others: Reserved

22.2.33 DIEPDMAB0: Device IN endpoint 16 buffer address register

DIEPDMAB0=0x5000_091C

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
DMABufferAddr			
0			
R			

31 0	DMABufferAddr	Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress.

22.2.34 DIEPCTLi : Device control IN endpoint i control register

For I = 1; I <= OTG_NUM_IN_EPS

DIEPCTLi =0x5000_0920

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
EPEna	EPDis	Reserved	SNAK	CNAK	TxFNum		Stall	Reserved	EPType	NAKSts	DPID	USBActEP	Reserved	MPS	
0	0	-	0	0	0		0	-	0	0	0	0	-	0	
RW	RW	-	W	W	RW		RW	-	R	R	R	R	-	RW	

31	EPEna	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled, -For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. -For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. • The core clears this bit before setting any of the following interrupts on this endpoint: -SETUP Phase Done -Endpoint Disabled -Transfer Completed Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. Values: • 0x1 (ACTIVE): Enable Endpoint • 0x0 (INACTIVE): No Action Value After Reset: 0x0
30	EPDis	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. Values: • 0x1 (ACTIVE): Disable Endpoint • 0x0 (INACTIVE): No Action Value After Reset: 0x0
27	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. Values: • 0x1 (ACTIVE): Set NAK • 0x0 (INACTIVE): No Set NAK Value After Reset: 0x0

26	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: • 0x1 (ACTIVE): Clear NAK • 0x0 (INACTIVE): No Clear NAK Value After Reset: 0x0
25 22	TxFNum	TxFIFO Number (TxFNum) Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. Values: • 0x3 (TXFIFO3): Tx FIFO 3 • 0x4 (TXFIFO4): Tx FIFO 4 • 0x0 (TXFIFO0): Tx FIFO 0 • 0x1 (TXFIFO1): Tx FIFO 1 • 0x2 (TXFIFO2): Tx FIFO 2 Value After Reset: 0x0
21	Stall	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (ACTIVE): STALL All Active Tokens • 0x0 (INACTIVE): STALL All non-active tokens Value After Reset: 0x0
19 18	EPType	Endpoint Type (EPType) This is the transfer type supported by this logical endpoint. • 2'b00: Control • 2'b01: Isochronous • 2'b10: Bulk • 2'b11: Interrupt Values: • 0x2 (BULK): Bulk • 0x3 (INTERRUP): Interrupt • 0x0 (CONTROL): Control • 0x1 (ISOCHRONOUS): Isochronous Value After Reset: 0x0

17	NAKSts	NAK Status (NAKSts) Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. • 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: • The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. • For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. • For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint • 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status Value After Reset: 0x0
16	DPID	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. • 1'b0: DATA0 • 1'b1: DATA1 Values: • 0x0 (DATA0EVENFRM): DATA0 or Even Frame • 0x1 (DATA1ODDFRM): DATA1 or Odd Frame Value After Reset: 0x0
15	USBActEP	USB Active Endpoint (USBActEP) Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. Values: • 0x0 (DISABLED): Not Active • 0x1 (ENABLED): USB Active Endpoint Value After Reset: 0x0
10 0	MPS	Maximum Packet Size (MPS) The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes. Value After Reset: 0x0

22.2.35 DIEPINTi: Device IN endpoint I interrupt register

For I = 1; I <= OTG_NUM_IN_EPS

DIEPINT0=0x5000_0928

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			NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn	TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEm	Reserved	AHBErr	EPDisbld	XferCompl
-			0	0	0	-		0	0	1	0	0	0	-	0	0	0
-			RW	RW	RW	-		RW	RW	R	RW	RW	RW	-	RW	RW	RW

14	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: • 0x1 (ACTIVE): NYET Interrupt • 0x0 (INACTIVE): No NYET interrupt Value After Reset: 0x0
13	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. Values: • 0x1 (ACTIVE): NAK Interrupt • 0x0 (INACTIVE): No NAK interrupt Value After Reset: 0x0
12	BbleErr	NAK Interrupt (BbleErr) The core generates this interrupt when babble is received for the endpoint. Values: • 0x1 (ACTIVE): BbleErr interrupt • 0x0 (INACTIVE): No interrupt Value After Reset: 0x0
9	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. Values: • 0x1 (ACTIVE): BNA interrupt • 0x0 (INACTIVE): No BNA interrupt Value After Reset: 0x0
8	TxfifoUndrn	Fifo Underrun (TxfifoUndrn) Applies to IN endpoints Only This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. Values: • 0x1 (ACTIVE): TxFIFO Underrun interrupt • 0x0 (INACTIVE): No Tx FIFO Underrun interrupt Value After Reset: 0x0

7	TxFEmp	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). Values: • 0x1 (ACTIVE): Transmit FIFO Empty interrupt • 0x0 (INACTIVE): No Transmit FIFO Empty interrupt Value After Reset: 0x1
6	INEPNakEff	IN Endpoint NAK Effective (INEPNakEff) Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. Values: • 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt • 0x0 (INACTIVE): No Endpoint NAK Effective interrupt Value After Reset: 0x0
5	INTknEPMis	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Values: • 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt • 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt Value After Reset: 0x0
4	INTknTXFEmp	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. Values: • 0x1 (ACTIVE): IN Token Received Interrupt • 0x0 (INACTIVE): No IN Token Received interrupt Value After Reset: 0x0
2	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. Values: • 0x1 (ACTIVE): AHB Error interrupt • 0x0 (INACTIVE): No AHB Error Interrupt Value After Reset: 0x0

1	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: • 0x1 (ACTIVE): Endpoint Disabled Interrupt • 0x0 (INACTIVE): No Endpoint Disabled Interrupt Value After Reset: 0x0
0	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled -For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. -For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. Values: • 0x1 (ACTIVE): Transfer Complete Interrupt • 0x0 (INACTIVE): No Transfer Complete Interrupt Value After Reset: 0x0

22.2.36 DIEPDMAi: Device IN endpoint I DMA address register

For I = 1; I <= OTG_NUM_IN_EPS

DIEPDMAi =0x5000_0934

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
DMAAddr			
0			
RW			

31 0	DMAAddr	Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. • When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. • When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

22.2.37 DTXFSTSi: Device IN endpoint transmit FIFO status register i

Device IN Endpoint Transmit FIFO Status Register i. Device IN Endpoint I DMA Address Register

DTXFSTSi =0x5000_0938

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		INEPTxFSpcAvail	
-		0x0040	
-		R	

15 0	INEPTxFSpcAvail	IN Endpoint TxFIFO Space Avail (INEPTxFSpcAvail) Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. • 16'h0: Endpoint TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16'hn: n words available (where 0 n 32,768) • 16'h8000: 32,768 words available • Others: Reserved

22.2.38 DIEPDMABi: Device IN endpoint I buffer address register

Device IN Endpoint I Buffer Address Register.

For I = 1; I <= OTG_NUM_IN_EPS

DIEPDMABi =0x5000_093C

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
DMABufferAddr			
0			
R			

31 0	DMABufferAddr	Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

22.2.39 DOEPCTL0: Device control OUT endpoint 0 control register

DOEPCTL0=0x5000_0B00

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
EPEna	EPDis	Reserved	SNAK	CNAK	Reserved		Stall	Snp	EPType	NAKSts	Reserved	USBActEP	Reserved		MPS
0	0	-	0	0	-		0	0	0	0	-	1	-		0
RW	RW	-	W	W	-		RW	RW	R	R	-	R	-		R

31	EPEna	Endpoint Enable (EPEna) • When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. • The core clears this bit before setting any of the following interrupts on this endpoint: -SETUP Phase Done -Endpoint Disabled -Transfer Completed Note: This bit must be set for the core to transfer SETUP data packets into memory. Values: • 0x1 (ACTIVE): Enable Endpoint • 0x0 (INACTIVE): No action Value After Reset: 0x0
30	EPDis	Endpoint Disable (EPDis) The application cannot disable control OUT endpoint 0. Values: • 0x0 (INACTIVE): No Endpoint disable Value After Reset: 0x0
27	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint. Values: • 0x0 (NOSET): No action • 0x1 (SET): Set NAK Value After Reset: 0x0
26	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: • 0x1 (CLEAR): Clear NAK • 0x0 (NOCLEAR): No action Value After Reset: 0x0
21	Stall	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is Set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (ACTIVE): Stall Handshake • 0x0 (INACTIVE): No Stall Value After Reset: 0x0

20	Snp	Snoop Mode (Snp) This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory. Values: • 0x0 (DISABLED): Snoop Mode disabled • 0x1 (ENABLED): Snoop Mode enabled Value After Reset: 0x0
19 18	EPType	Endpoint Type (EPType) Hardcoded to 2'b00 for control. Values: • 0x0 (ACTIVE): Endpoint Control 0 Value After Reset: 0x0
17	NAKSts	NAK Status (NAKSts) Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. • 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even If there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint • 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status Value After Reset: 0x0
15	USBActEP	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces. Values: • 0x1 (ACTIVE): USB Active Endpoint 0 Value After Reset: 0x1
1 0	MPS	Maximum Packet Size (MPS) The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. • 2'b00: 64 bytes • 2'b01: 32 bytes • 2'b10: 16 bytes • 2'b11: 8 bytes Values: • 0x2 (BYTE16): 16 bytes • 0x3 (BYTE8): 8 bytes • 0x1 (BYTE32): 32 bytes • 0x0 (BYTE64): 64 bytes Value After Reset: 0x0

22.2.40 DOEPINT0: Device OUT endpoint 0 interrupt register

DOEPINT0=0x5000_0B08

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			NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	OutPktErr	Reserved	Back2BackS	StsPhseRcvd	OUTTknEPdi	SetUp	AHBErr	EPDisbld	XferCompl
-			0	0	0	-		0	0	-	0	0	0	0	0	0	0
-			RW	RW	RW	-		RW	RW	-	RW	RW	RW	RW	RW	RW	RW

14	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: • 0x1 (ACTIVE): NYET Interrupt • 0x0 (INACTIVE): No NYET interrupt Value After Reset: 0x0
13	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. Values: • 0x1 (ACTIVE): NAK Interrupt • 0x0 (INACTIVE): No NAK interrupt Value After Reset: 0x0
12	BbleErr	NAK Interrupt (BbleErr) The core generates this interrupt when babble is received for the endpoint. Values: • 0x1 (ACTIVE): BbleErr interrupt • 0x0 (INACTIVE): No BbleErr interrupt Value After Reset: 0x0
9	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) The core generates this interrupt when the descriptor accessed is not ready for the core to process, such as Host busy or DMA done. Values: • 0x1 (ACTIVE): BNA interrupt • 0x0 (INACTIVE): No BNA interrupt Value After Reset: 0x0
8	OutPktErr	OUT Packet Error (OutPktErr) Applies to OUT endpoints Only This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. Values: • 0x1 (ACTIVE): OUT Packet Error • 0x0 (INACTIVE): No OUT Packet Error Value After Reset: 0x0
6	Back2BackSETup	Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, Values: • 0x1 (ACTIVE): Back-to-Back SETUP Packets Received • 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received Value After Reset: 0x0

5	StsPhseRcvd	Status Phase Received for Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. Values: • 0x1 (ACTIVE): Status Phase Received for Control Write • 0x0 (INACTIVE): No Status Phase Received for Control Write Value After Reset: 0x0
4	OUTTknEPdis	OUT Token Received When Endpoint Disabled (OUTTknEPdis) Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. Values: • 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled • 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled Value After Reset: 0x0
3	SetUp	SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. Values: • 0x1 (ACTIVE): SETUP Phase Done • 0x0 (INACTIVE): No SETUP Phase Done Value After Reset: 0x0
2	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. Values: • 0x1 (ACTIVE): AHB Error interrupt • 0x0 (INACTIVE): No AHB Error Interrupt Value After Reset: 0x0
1	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: • 0x1 (ACTIVE): Endpoint Disabled Interrupt • 0x0 (INACTIVE): No Endpoint Disabled Interrupt Value After Reset: 0x0
0	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. • For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. • For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. Values: • 0x1 (ACTIVE): Transfer Complete Interrupt • 0x0 (INACTIVE): No Transfer Complete Interrupt Value After Reset: 0x0

22.2.41 DOEPDMA0: Device OUT endpoint 0 DMA address register

DOEPDMA0=0x5000_0B14

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
DMAAddr			
0			
RW			

31 0	DMAAddr	Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

22.2.42 DOEPDMAB0: Device OUT Endpoint 16 Buffer Address Register

DOEPDMAB0=0x5000_0B1C

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
DMABufferAddr			
0			
R			

31 0	DMABufferAddr	Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

22.2.43 DOEPCTLi: Device control OUT endpoint I control register

Device Control OUT Endpoint I Control Register.

For I = 1; I <= OTG_NUM_IN_EPS

DOEPCTLi =0x5000_0B20

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
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved		Stall	Snp	EPType	NAKSts	DPID	USBActEP	Reserved	MPS	
0	0	0	0	0	0	-		0	0	0	0	0	0	-	0	
RW	RW	W	W	W	W	-		RW	RW	R	R	R	R	-	RW	

31	EPEna	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. • For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. • For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: • SETUP Phase Done • Endpoint Disabled • Transfer Completed Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. Values: • 0x1 (ACTIVE): Enable Endpoint • 0x0 (INACTIVE): No Action Value After Reset: 0x0
30	EPDis	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. Values: • 0x1 (ACTIVE): Disable Endpoint • 0x0 (INACTIVE): No Action Value After Reset: 0x0
29	SetD1PID	Set DATA1 PID (SetD1PID) • Applies to interrupt and bulk IN and OUT endpoints only. • Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. Values: • 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame • 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)frame Value After Reset: 0x0

28	SetD0PID	Set DATA0 PID (SetD0PID) • Applies to interrupt/bulk IN and OUT endpoints only. • Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. Values: • 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame • 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)frame Value After Reset: 0x0
27	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. Values: • 0x1 (ACTIVE): Set NAK • 0x0 (INACTIVE): No Set NAK Value After Reset: 0x0
26	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: • 0x1 (ACTIVE): Clear NAK • 0x0 (INACTIVE): No Clear NAK Value After Reset: 0x0
21	Stall	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (ACTIVE): STALL All Active Tokens • 0x0 (INACTIVE): STALL All non-active tokens Value After Reset: 0x0
20	Snp	Snoop Mode (Snp) Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory. Values: • 0x0 (DISABLE): Disable Snoop Mode • 0x1 (ENABLE): Enable Snoop Mode Value After Reset: 0x0
19 18	EPType	Endpoint Type (EPType) This is the transfer type supported by this logical endpoint. • 2'b00: Control • 2'b01: Isochronous • 2'b10: Bulk • 2'b11: Interrupt Values: • 0x2 (BULK): Bulk • 0x3 (INTERRUPT): Interrupt • 0x0 (CONTROL): Control • 0x1 (ISOCHRONOUS): Isochronous Value After Reset: 0x0

17	NAKSts	NAK Status (NAKSts) Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. • 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: • The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. • For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. • For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. Values: • 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint • 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status Value After Reset: 0x0
16	DPID	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. • 1'b0: DATA0 • 1'b1: DATA1 Values: • 0x1 (ACTIVE): Endpoint Data PID active • 0x0 (INACTIVE): Endpoint Data PID not active Value After Reset: 0x0
15	USBActEP	USB Active Endpoint (USBActEP) Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. Values: • 0x0 (DISABLED): Not Active • 0x1 (ENABLED): USB Active Endpoint Value After Reset: 0x0
10 0	MPS	Maximum Packet Size (MPS) The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes. Value After Reset: 0x0

22.2.44 DOEPINTi: Device OUT endpoint I interrupt register

For I = 1; I <= OTG_NUM_IN_EPS

DOEPINTi =0x5000_0B28

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			NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr	Reserved	Back2BackS	StsPhseRcvd	OUTTknEPdi	SetUp	AHBErr	EPDisbld	XferCompl
-			0	0	0	0	-	0	0	-	0	0	0	0	0	0	0
-			RW	RW	RW	RW	-	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

14	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: • 0x1 (ACTIVE): NYET Interrupt • 0x0 (INACTIVE): No NYET interrupt Value After Reset: 0x0
13	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. Values: • 0x1 (ACTIVE): NAK Interrupt • 0x0 (INACTIVE): No NAK interrupt Value After Reset: 0x0
12	BbleErr	NAK Interrupt (BbleErr) The core generates this interrupt when babble is received for the endpoint. Values: • 0x1 (ACTIVE): BbleErr interrupt • 0x0 (INACTIVE): No BbleErr interrupt Value After Reset: 0x0
11	PktDrpSts	Packet Drop Status (PktDrpSts) This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected. Values: • 0x1 (ACTIVE): Packet Drop Status interrupt • 0x0 (INACTIVE): No interrupt Value After Reset: 0x0
9	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done Values: • 0x1 (ACTIVE): BNA interrupt • 0x0 (INACTIVE): No BNA interrupt Value After Reset: 0x0

8	OutPktErr	OUT Packet Error (OutPktErr) Applies to OUT endpoints Only This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. Values: • 0x1 (ACTIVE): OUT Packet Error • 0x0 (INACTIVE): No OUT Packet Error Value After Reset: 0x0
6	Back2BackSETup	Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, Values: • 0x1 (ACTIVE): Back-to-Back SETUP Packets Received • 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received Value After Reset: 0x0
5	StsPhseRcvd	Status Phase Received for Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. Values: • 0x1 (ACTIVE): Status Phase Received for Control Write • 0x0 (INACTIVE): No Status Phase Received for Control Write Value After Reset: 0x0
4	OUTTknEPdis	OUT Token Received When Endpoint Disabled (OUTTknEPdis) Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. Values: • 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled • 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled Value After Reset: 0x0
3	SetUp	SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. Values: • 0x1 (ACTIVE): SETUP Phase Done • 0x0 (INACTIVE): No SETUP Phase Done Value After Reset: 0x0
2	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. Values: • 0x1 (ACTIVE): AHB Error interrupt • 0x0 (INACTIVE): No AHB Error Interrupt Value After Reset: 0x0

1	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: • 0x1 (ACTIVE): Endpoint Disabled Interrupt • 0x0 (INACTIVE): No Endpoint Disabled Interrupt Value After Reset: 0x0
0	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled -For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. -For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. Values: • 0x1 (ACTIVE): Transfer Complete Interrupt • 0x0 (INACTIVE): No Transfer Complete Interrupt Value After Reset: 0x0

22.2.45 DOEPDMAi: Device OUT Endpoint I DMA Address Register

For I = 1; I <= OTG_NUM_IN_EPS

DOEPDMAi =0x5000_0B34

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
DMAAddr			
0			
RW			

31 0	DMAAddr	Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. • When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

22.2.46 DOEPDMABi: Device OUT Endpoint I Buffer Address Register

For I = 1; I <= OTG_NUM_IN_EPS

DOEPDMABi =0x5000_0B3C

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
DMABufferAddr			
0			
R			

31 0	DMABufferAddr	Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress.

22.2.47 PCGCCTL: Power and Clock Gating Control Register

PCGCCTL =0x5000_0E00

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
RestoreValue			EssRegRestored	Reserved	RestoreMode	Reserved	L1Suspended	PhySleep	Enbl_L1Gatin	Reserved	RstPdwnMod	PwrClmp	GateHclk	StopPclk
0			0	-	0	-	0	0	0	-	0	0	0	0
RW			RW	-	RW	-	R	R	RW	-	RW	RW	RW	RW

31 14	RestoreValue	Restore Value (RestoreValue) When Hibernation mode is enabled, the RestoreValue needs to be read at SAVE_POINT to store in non-volatile memory and at RESTORE_POINT restored before PCGCCTL.EssRegRestored field is set. • [31] if_dev_mode -1: core restored as device • [30:29] p2hd_prt_spd (PRT speed) -01: FS -10: LS -11: Reserved • [28:27] p2hd_dev_enum_spd (Device enumerated speed) -10: LS -11: FS (48 MHz clk) • [26:20] mac_dev_addr (MAC device address) Device address • [19] mac_termselect (Termination selection) -1: FS_TERM (Program for Full Speed) • [18:17] mac_xcvrselect (Transceiver select) -01: FS_XCVR (Full Speed) -10: LS_XCVR (Low Speed) -11: LFS_XCVR (Reserved) • [15:14] prt_clk_sel (Refer prt_clk_sel table) Defines port clock select for different speeds.

13	EssRegRestored	Essential Register Values Restored (EssRegRestored) A write of one into this field indicates that register values of essential registers have been restored. For definition of essential registers, refer programming flow section for Hibernation. Values: • 0x0 (NOT_RESTORED): Register values of essential registers are not restored • 0x1 (RESTORED): Register values of essential registers have been restored
9	RestoreMode	Restore Mode (RestoreMode) The application should program this bit to specify the restore mode during RESTORE POINT before programming PCGCCTL.EssRegRest bit is set. • Device Mode: -1'b0: Device-initiated Remote Wakeup -1'b1: Host-initiated Resume, Host-initiated Reset Values: • 0x0 (DISABLED): In Device mode, this bit indicates Device-initiated Remote Wakeup • 0x1 (ENABLED): In Device mode, this bit indicates Host-initiated Resume and Reset
7	L1Suspended	L1 Deep Sleep Indicates that the PHY is in deep sleep when in L1 state. Values: • 0x1 (ACTIVE): Deep Sleep • 0x0 (INACTIVE): Non Deep Sleep
6	PhySleep	PHY In Sleep Indicates that the PHY is in Sleep State. Values: • 0x1 (ACTIVE): Phy in Sleep state • 0x0 (INACTIVE): Phy not in Sleep state
5	Enbl_L1Gating	Enable Sleep Clock Gating If this bit is set, core internal clock gating is enabled sleep state if utmi_l1_suspend_n cannot be asserted by the core. The PHY clock will not be gated in sleep state if Enbl_L1Gating is not set. Values: • 0x0 (DISABLED): The PHY clock is not gated in Sleep state • 0x1 (ENABLED): The Core internal clock gating is enabled in Sleep state
3	RstPdwnModule	Reset Power-Down Modules (RstPdwnModule) This bit is valid only in Partial Power-Down mode. • The application sets this bit when the power is turned off. • The application clears this bit after the power is turned on and the PHY clock is up. Note: The R/W of all core registers are possible only when this bit is set to 1b0. Values: • 0x1 (OFF): Power is turned off • 0x0 (ON): Power is turned on
2	PwrClmp	Power Clamp (PwrClmp) This bit is valid only in Partial Power-Down mode • The application sets this bit before the power is turned off to clamp the signals between the power-on modules and the power-off modules. • The application clears the bit to disable the clamping before the power is turned on. Values: • 0x0 (DISABLED): Clears this bit to disable the clamping before the power is turned on • 0x1 (ENABLED): In only Partial Power-Down mode, sets this bit to clamp the signals between the power-on modules and the power-off modules before the power is turned off

1	GateHclk	Gate Hclk (GateHclk) • The application sets this bit to gate hclk to modules other than the AHB Slave and Master and wakeup logic when the USB is suspended or the session is not valid. • The application clears this bit when the USB is resumed or a new session starts. Values: • 0x0 (DISABLED): Clears this bit when the USB is resumed or a new session starts • 0x1 (ENABLED): Sets this bit to gate hclk to modules when the USB is suspended or the session is not valid
0	StopPclk	Stop Pclk (StopPclk) • The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. • The application clears this bit when the USB is resumed or a new session starts. Values: • 0x0 (DISABLED): Disable Stop Pclk • 0x1 (ENABLED): Enable Stop Pclk

22.3 Functional description

22.3.1 USB device

USB devices are peripherals communicated to a USB Host. Devices are implemented by adding USB Software Stack. USB Device peripherals can have one or more of the following USB Device Classes.

- Human Interface Device (HID) is typically used for devices such as keyboards, mouse, or joysticks.
- Communication Device Class (CDC) provides virtual communication port functionality to the USB Host.
- Mass Storage Class (MSC) is used to allow access to various storage devices to the USB Host.
- Custom Class provides a vendor specific USB Device Class.
- For more information about USB Device Classes, it can be found on USB-IF homepage.

USB Host initiates all data of USB Communication. USB consist of one or more transactions. The data sends and receives from Endpoints.

Starting transaction, USB need to make Pipes between Device and Host logically. Host sends data from OUT Endpoint. Host receives data from IN Endpoint.

An endpoint acts as a kind of buffer. It can be configured to support four transfer types defined in the USB specification.

- Control Transfer is typically used to setup an USB device. It always uses IN/OUT Endpoint 0.
- Interrupt Transfer can be used where data is sent regularly, for example for status updates.
- Isochronous Transfer transmits real-time data such as audio and video. It has a guaranteed, fixed bandwidth, but no error detection.
- Bulk Transfer can be used to send data where timing is not important, for example to a printer.

Data is transferred in so called transactions. Normally, they consist of three packets:

- Token packet is the header defining the transaction type and direction, the device address, and the endpoint.
- Data is transferred in a data packet.
- Final status of the transaction is acknowledges in the handshake packet.

USB packet format is describes in Figure 153.

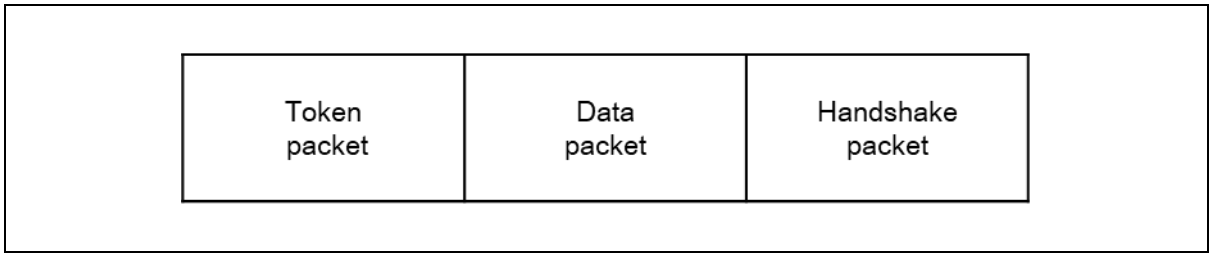


Figure 153. USB Packet Field

Table 87. USB Packet PID

Token packets		
OUT	0001b	Beginning of OUT transaction
IN	1001b	Start of an IN transaction
SETUP	0101b	Signals beginning of a Control Transaction
SOF	1101b	Start of Frame packet
Data packets		
DATA0	0010b	Clears the toggle bit in the data packet
DATA1	1011b	Sets the toggle bit in the data packet
Handshake		
ACK	0110b	Transmission successfully completed
NACK	1010b	Transmission unsuccessful
STALL	1110b	delayed response, retry needed

22.3.2 USB descriptors

USB devices report their attributes by using descriptors, which are data structures with a defined format.

When a USB device is attached to the USB bus, a host uses a process known as bus enumeration to identify and configure the device. If the enumeration is finished, the device can service the active endpoints to exchange data with the USB Host.

Following is the most commonly used descriptors:

- Device Descriptor
- Configuration Descriptor
- Interface Descriptor
- Endpoint Descriptor
- String Descriptor

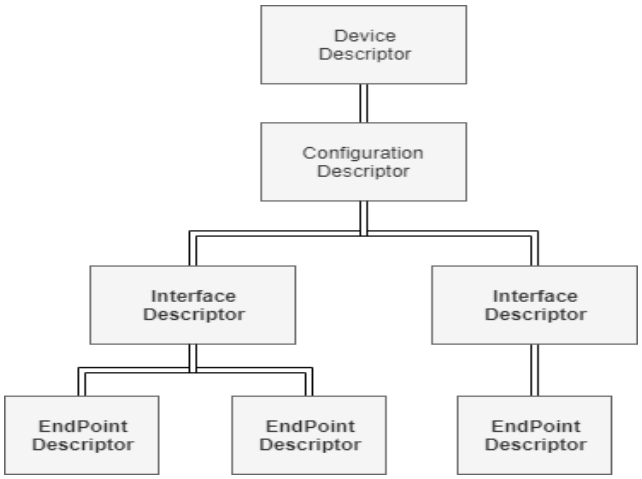


Figure 154. USB Descriptor Hierarchy

22.3.3 USB operation

Reliable operation of USB requires the followings:

1. USB power system: There are 2 way of supplying USB power.
   - First, integrated 3.3V LDO can be supplied to USB module. SCU_VDCCON register is set to 0xA3000000.
   - Second, 3.3V VDDIO2 can be supplied to USB module. SCU_VDCCON register is set to 0xAA00200 and 21bit of PCU_USBCON register is set to 1.
2. USB clock speed: 48MHz clock speed is required for full speed USB. For reliable communication, USB recommends to use the clock which is a tolerance of ±2500 ppm(0.25%). For more information, please refer to Chapter 4.2 Clock System.
3. USB port control (D+, D-): A user needs to configure PCU_USBCON register for corresponding USB function. The register regards to characteristics of USB port such as pull up/down and drive strength option.
   - Example code for the device using full speed USB: PCU_USBCON will be set as shown below:

```
USBCON =    (0x7 << 28) | (0x7 << 24)                          //Drive Strengthen
            | (0x1 << 23) | (0x0 << 22) | (0x1 << 21)          //CONTROL_EN//PHY_SUSPEND//VREF_SEL
            | (0x0 << 19) | (0x0 << 18)                        //PD_ENDN, DP
            | (0x0 << 17) | (0x1 << 16)                        //PU_ENDN, DP
            | (0x0 << 12) | (0x0 << 8)          //VIH_CON//VIL_CON
            | (0x7 << 4) | (0x7 << 0);          //PD_CON//PU_CON
```

4. Device initialization, connection, and disconnection: The application must perform the following steps to initialize the controller at a device.
   - Program the following fields in DCFG register.
     i. DescDMA bit
     ii. Device Speed
     iii. NonZero Length Status OUT Handshake
     iv. Periodic Frame Interval (If Periodic Endpoints are supported)
   - Program the Device threshold control register. This is required only if you are planning to enable thresholding.
   - Clear the DCTL.SftDiscon bit. The controller issues a connection after this bit is cleared.

- Program the GINTMSK register to unmask the following interrupts.
  i. USB Reset
  ii. Enumeration Done
  iii. Early Suspend
  iv. USB SuspenD
  v. SOF
- Wait for the GINTSTS.USBReset interrupt, which indicates a reset has been detected on the USB and lasts for about 10 ms. Initialization on USB Reset.
  i. DOEPCTLn.SNAK = 1 (for all OUT endpoints)
  ii. DAINTMSK.INEP0 = 1 (control 0 IN endpoint)
  iii. DAINTMSK.OUTEP0 = 1 (control 0 OUT endpoint)
  iv. DOEPMSK.SETUP = 1
  v. DOEPMSK.XferCompl = 1
  vi. DIEPMSK.XferCompl = 1
  vii. DIEPMSK.TimeOut = 1
- Reset the Device Address field in Device Configuration Register (DCFG).
  i. USB->DCFG &= ~(0x7f << 4); //clear
- Wait for the GINTSTS.EnumerationDone interrupt. This interrupt indicates the end of reset on the USB. Initialization on Enumeration Completion.
  i. On the Enumeration done interrupt (GINTSTS.EnumDone, read the DSTS register to determine the enumeration speed.
  ii. Program the DIEPCTL0.MPS field to set the maximum packet size. This step configures control endpoint 0. The maximum packet size for a control endpoint depends on the enumeration speed.
  iii. The descriptors must be set up in memory before enabling the endpoint.
    - DOEPCTL0.EPEna = 1
  iv. Unmask the SOF interrupt.

If VBUS is on when the device is connected to the USB cable, the device connection flow is as follows:

1. The device triggers the GINTSTS.SessReqInt [bit 30] interrupt bit.
2. When the device application detects the GINTSTS.SessReqInt interrupt, it programs the required bits in the DCFG register.
3. When the Host drives Reset, the Device triggers GINTSTS.USBRst [bit 12] on detecting the Reset.

The device session ends when the USB cable is disconnected or if the VBUS is switched off by the Host. The device disconnect flow is as follows:

1. When the USB cable is unplugged or when the VBUS is switched off by the Host, the Device controller triggers GINTSTS.USBRst [bit 12] interrupt bit.
2. When the device application detects GINTSTS.USBRst, the application sets a time-out check for SET ADDRESS Control Xfer from Host.
3. If application does not receive SET ADDRESS Control Xfer from Host before the time-out period, it is treated as a device disconnection.

22.3.4 USB transfer

In this section, a sequence that an application must do when it receives a command from Host.

The application must do the followings when it receives a SetAddress command in a SETUP packet:

1. Program the DCFG register with the device address received in the SetAddress command.
2. Program the controller to send out a status IN packet.

The application must do the followings when it receives a SetConfiguration or SetInterface command in a SETUP packet.

1. The application must program the endpoint registers of the endpoints affected by this command.
2. After all required endpoints are configured, the application must program the controller to send a status IN packet.

Following procedure describes how the application initializes the channel and provides information on transfers IN/OUT EndPoint. Data buffers are presented through descriptor structures:

1. The application prepares the descriptors, and sets the DIEPCTLn/DOEPCTLn.EPEna bit.
2. The DMA fetches the corresponding descriptor (initially determined by DIEPDMAn/DOEPDMAn).
3. The DMA internally sets the transfer size from descriptor back to DIEPTSIZn/DOEPTSIZn.

4. From this point onwards, the current device flow executes.
5. Once the transfer size data is moved by the DMA, the DMA checks for further links in the descriptor chain.
6. If this is the last descriptor, the DMA sets the DIOEPINTn.XferCompl interrupt.

22.3.5 USB software example

This section describes how to implement USB software and how to use USB application between host and a device.

1) CDC(Virtual Comport)

Assuming that CDC (Communications Device Class) provides the developer with a simple data exchange interface, as a brief example of this interface, a simple application can be introduced in the followings. It (ACM class) results in a virtual COM port on the host.

Refer to VCP (Virtual COM port) example project. This application sends and receives a string over USB to a virtual COM port on the host. Terminal emulator program such as Tera Term and Putty will be required for users.

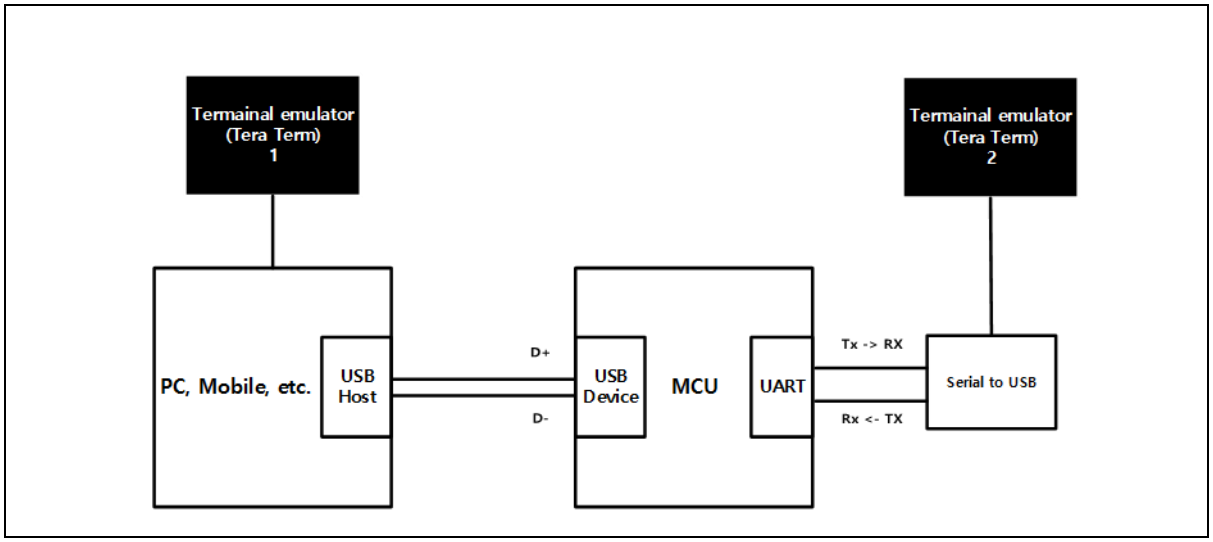


Figure 155. USB Communication using Software

<u>Application example</u>

1. Open USB VCP test example project. Compile and download the project.
2. PC (Host) USB connect to MCU (Device) USB port.
3. After PC (Host) recognize USB Device, user need to set up Com Port Driver (inf file).

4. Open Windows OS – Device Manager console

Figure 156 Application Example: Open Windows OS

5. Update manually the driver software. ABOV provides G324_Comport_Driver.inf file.

6. Check the Com port to recognize USB Device properly.

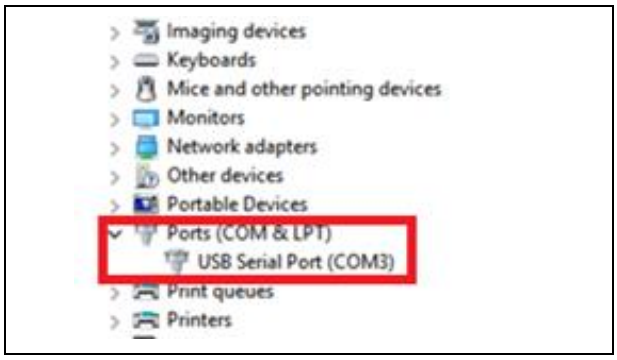


Figure 157. Application Example: Check COM Port

7. Open Tera Term for Serial to USB (UART) and USB.

8. After setting the connection, the user sends a character from USB to UART. Tera Term (UART) prints the character after receiving. Conversely, the user sends a character from UART to USB. Tera Term (USB) prints the character.

9. Following figures are examples, printing a string "ABOV From xxx DATA".

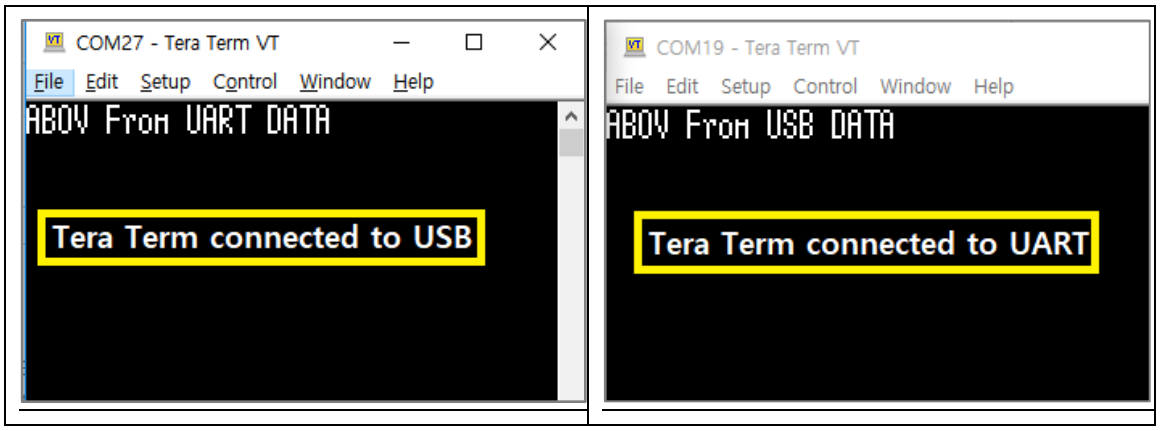


Figure 158. Application Example: Printing Characters

2) HID(Custom Data-pipe)

It is possible to use custom data exchange based on the HID class.

Refer to HID(HID_Custom) example project. This application sends and receive a string over USB to a HID GUI tool(Including USB example project –> PC_Tool folder) on the host.

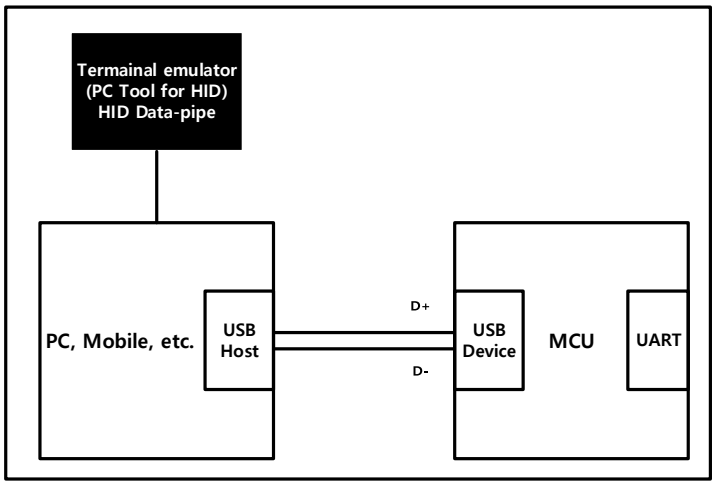


Figure 159. HID Block Diagram

A. Open USB VCP test example project. Complie and download the project.

B. PC(Host) USB connect to MCU(Device) usb port.

C. Open Windows OS – Device Manager console

D. Check the HID device to recognize USB Device properly.

E. Open HID PC GUI Tool for communicating via USB.

F. As shown in the Figure 160 below. (For example, receive a string "0x55 0xAA…")

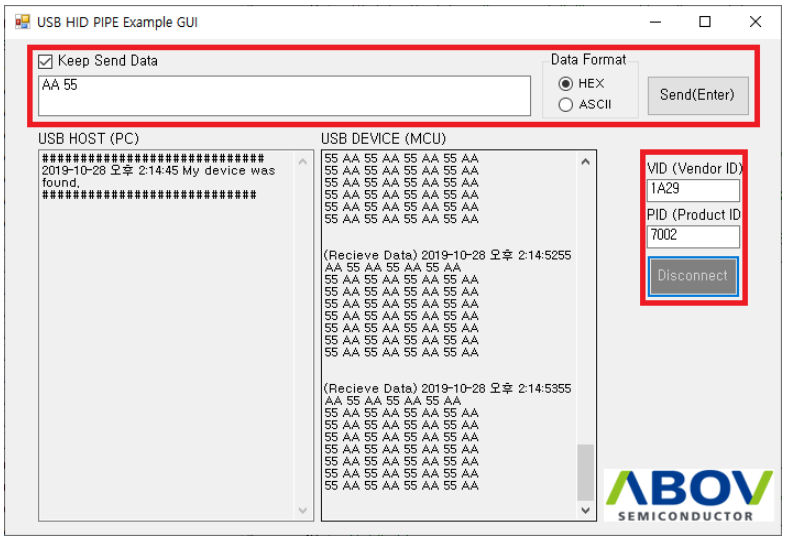


Figure 160. USB HID PIPE Example GUI

■ **Hardware**

As shown in a detailed reference design.

1. D+, D- line : 33Ω
2. VDDIO2 : There are 2 options.

   Internal USB power(VDD33) -> Floating, or External USB power -> supply 3.3V
3. For reliable communication, the USB specification requires the clock have a tolerance of ±2500 ppm.

   In order to meet this condition, It is recommended to use a crystal and PLL.

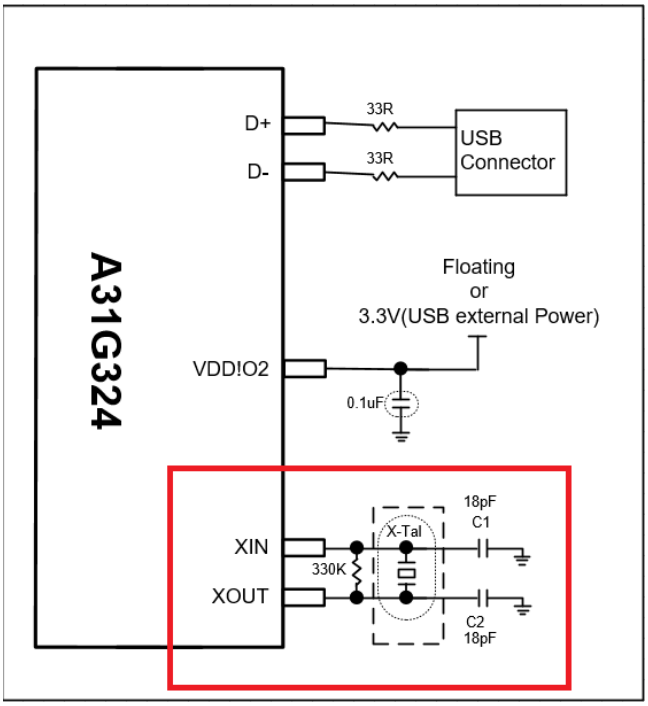


Figure 161. Crystal Design Guide

23. Real-time clock (RTC)

Real-time clock (RTC) of A31G32x series provides an automatic wakeup function to manage all low-power modes.

The RTC is an independent BCD timer/counter. It provides a time-of-day clock/calendar with programmable alarm interrupt. In addition, it includes a periodic programmable wakeup flag with interrupt capability. Eight RTC counters contain the sub-seconds, seconds, minutes, hours, week, day, month, and year expressed in binary coded decimal format (BCD). Compensations for 28-, 29- (leap year), 30-, and 31-day months are performed automatically. An error correction is available to compensate for variations in crystal oscillator accuracy.

In any mode of operation, the sub-second counter is never accessible. In normal operating mode, seven RTC counters are protected from read and write access. Read and write operation is possible only in RW mode. As long as the supply voltage remains in the operating range, the RTC never stops, regardless of the device status (run mode, low-power mode or under reset).

The RTC features the followings:

- LSI40KHz and MCCR5 clock as RTC clock (fRTC)
- Counters of year, month, week, day, hour, minute, and second (up to 99 years)
- Constant-period matching interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Timestamp interrupt and function
- Pin output of 1Hz clock
- Error correction by 1.53ppm resolution when using 32768Hz LSE
- Backup power mode operation support
- Four 32-bit backup registers (BKUR1~4) implemented in the RTC domain (remaining powered-on by VBAT when the VDD power is switched off)

Internal clock frequency for the RTC clock is detailed in electrical characteristic.

23.1 RTC block diagram

In this section, RTC block diagram is introduced in Figure 162.

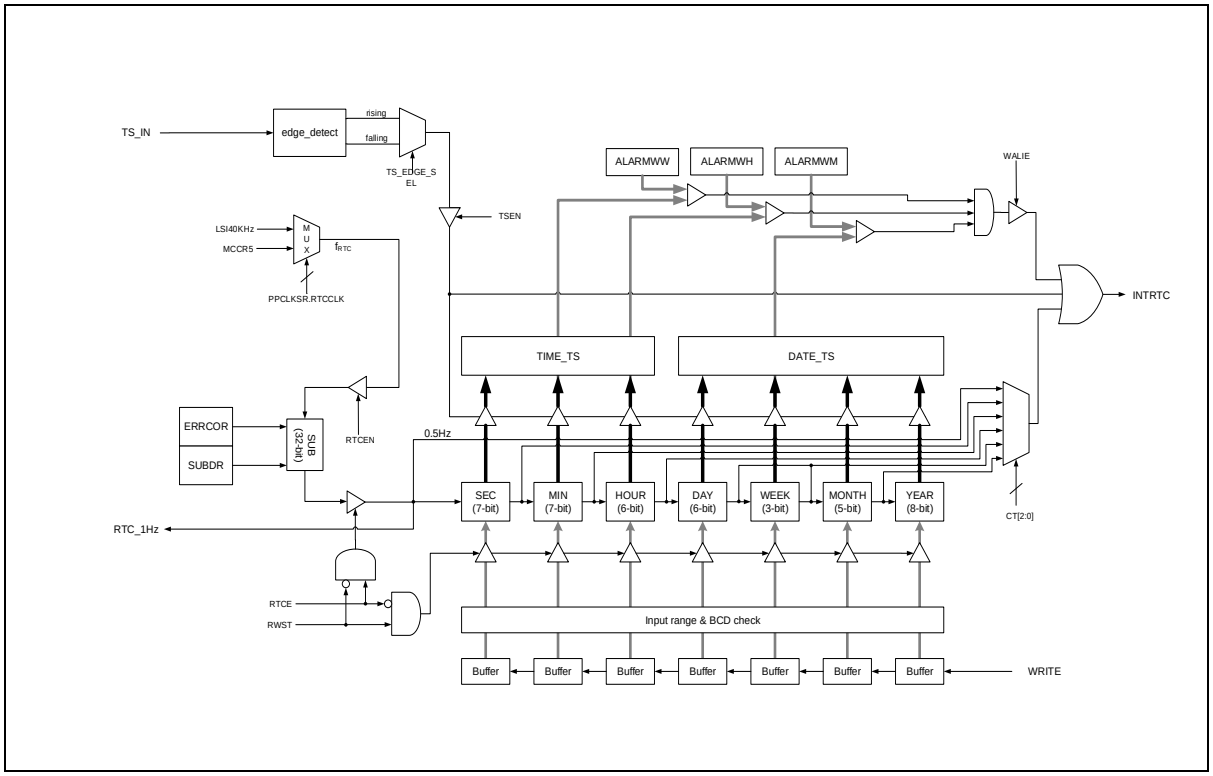


Figure 162. RTC Block Diagram

23.2 Registers

Base address of RTC block is introduced in the followings:

Table 88. Base Address of RTC Interface

Name	Base address
RTC	0x4000_6100

Table 89. RTC Register Map

Name	Offset	Type	Description	Reset value	Reference
RTCCR	0x0000	RW	RTC Control Register	0x0000_0000	23.2.1
ERRCOR	0x0008	RW	Watch Error Correction Register	0x0000_0000	23.2.2
SEC	0x000C	RW	Second Counter Register	0x0000_0000	23.2.3
MIN	0x0010	RW	Minute Counter Register	0x0000_0000	23.2.4
HOUR	0x0014	RW	Hour Counter Register	0x0012_0000	23.2.5
DAY	0x0018	RW	Day Counter Register	0x0001_0000	23.2.6
WEEK	0x001C	RW	Week Counter Register	0x0000_0000	23.2.7
MONTH	0x0020	RW	Month Counter Register	0x0001_0000	23.2.8
YEAR	0x0024	RW	Year Counter Register	0x0000_0000	23.2.9
ALARMW	0x0028	RW	Alarm Minute Register	0x0000_0000	23.2.10
SUBDR	0x002C	RW	Sub-second counter compare register	0x0001-FFFF	23.2.11
TIME_TS	0x0030	RW	TimeStamp time register	0x0000_0000	23.2.12
DATE_TS	0x0034	RW	TimeStamp date register	0x0000_0000	23.2.13
BKUR1	0x0038	RW	Backup register 1	0x0000_0000	23.2.14
BKUR2	0x003C	RW	Backup register 2	0x0000_0000	
BKUR3	0x0040	RW	Backup register 3	0x0000_0000	
BKUR4	0x0044	RW	Backup register 4	0x0000_0000	
RTCPFCR	0x0048	RW	RTC Pin Function Configuration register	0x0000_0000	23.2.15

23.2.1 RTCCR: RTC control register

RTCCR is a 32-bit register for controlling the RTC. You can set the alarm, constant-period matching, timestamp, etc. before activating the RTC. Changing this register during RTC operation may cause malfunction, so be sure to change while RTCE=0.

RTCCR=0x4000_6100

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
RTCEN	RTCE	Reserved		AMPM	CT	RIFG	TSEN	TS_EDGE_SEL	TSFG	WAIE	Reserved	WAFG	Reserved	RWST	RWAIT
0	0	-		0	0	0	0	0	0	0	-	0	-	0	0
RW	RW	-		RW	RW	RW	RW	RW	RW	RW		RW	-	RW	RW

31	RTCEN	RTC clock input enable	
		0	RTC clock input disabled
		1	RTC clock input enabled

NOTE:

If RTCEN = 0, writing to RTC registers is ignored. If the registers are read, only default values are read. (SEC to YEAR, ALARMWx, ERRCOR)

30	RTCE	RTC counter run	
		0	RTC counter stop
		1	RTC counter run
20	AMPM	Hour system selection	
		0	12-hour system (a.m. and p.m. are displayed.)
		1	24-hour system

NOTE:

AMPM can only be changed in RW mode. HOUR value for the 12-hour system and the 24-hour system are shown in Table 90.

19 17	CT	Period selection of constant-period matching	
		000	Constant-period matching disabled
		001	Once per 0.5 s (synchronized with second count up)
		010	Once per 1 s (same time as second count up)
		011	Once per 1 m (second 0 of every minute)
		100	Once per 1 hour (minute 0 and second 0 of every hour)
		101	Once per 1 day (hour 0, minute 0, and second 0 of every day)
		110	Once per 1 month (Day 1, hour 0 a.m., minute 0, and second 0 of every month)
		111	reserved
16	RIFG	Constant-period matching flag	
		0	Constant-period matching is not generated
		1	Constant-period matching is generated

NOTE:

This flag is reset when "0" is written to it. Writing "1" to it is invalid.

15	TSEN	Timestamp enable	
		0	Timestamp disabled
		1	Timestamp enabled

NOTE:

If timestamp wakeup is used in standby mode, PC13 must be set to timestamp function in run mode.

14 13	TS_EDGE_SEL	Timestamp event active edge selection	
		00	Falling edge
		01	Rising edge
		10	Falling/rising
		11	Reserved
12	TSFG	Time stamp event flag	
		0	Timestamp event is not generated
		1	Timestamp event is generated

NOTE:

This flag is reset when "0" is written to it. Writing "1" to it is invalid.

11	WAIE	Alarm enable	
		0	Alarm is disabled
		1	Alarm is enabled
8	WAFG	Alarm flag	
		0	Alarm is not generated
		1	Alarm is generated

NOTE:

This flag is reset when "0" is written to it. Writing "1" to it is invalid.

1	RWST	RW mode status flag	
		0	Run mode
		1	RW mode

NOTE:

WST is a signal that synchronizes RWAIT to RTC clock (f_{RTC}). This indicates that it is in the RW mode for reading and writing the RTC counter. Before reading and writing the RTC counter, confirm that the value of this flag is "1".

0	RWAIT	RW mode set	
		0	RW mode is unset
		1	RW mode is set

NOTES:

1. Constant-period matching interrupts, alarm match interrupts and timestamp interrupts use the same interrupt source.
2. When using these three types of interrupts at the same time, which interrupt occurred can be judged by checking the Constant-period matching flag (RIFG), the alarm detection status flag (WAFG), and the timestamp event flag (TSFG) upon RTC interrupt occurrence.
3. 25us delay required after RTCEN enable.
   Example code)
   RTCCR =| 0x01 (RTCEN enable)
   NOP; … NOP; // 25us

23.2.2 ERRCOR: Error correction register

ERRCOR is a 32-bit register. This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the sub-second counter register to the second counter register (SEC).

ERRCOR=0x4000_6108

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
DEV	DIR	Reserved	COREN	Reserved		CORDATA	
0	0	-	0	-		0x00000	
RW	RW	-	RW	-		RW	

31	DEV	Setting of watch error correction timing.	
		0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
		1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
30	DIR	Setting of watch error correction direction	
		0	Decreases by CORDATA.
		1	Increases by CORDATA.
28	COREN	Enable watch error correction	
		0	Disable
		1	Enable
16 0	CORDATA	Setting of watch error correction data When CORDATA=0, the watch error is not corrected.	

23.2.3 SEC: Second counter register

SEC is a 7-bit register that takes a value of '00' to '59' (decimal) and indicates the count value of seconds. It counts up when the sub-second counter register overflows. When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'.

Set a decimal value of '00' to '59' to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is written.

SEC =0x4000_610C

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			SEC
-			0x00
-			RW

6 0	SEC	Second Counter Register

23.2.4 MIN: Minute counter register

MIN is a 7-bit register that takes a value of '00' to '59' (decimal) and indicates the count value of minutes. It counts up when the second counter overflows. When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the second counter register overflows while this register is being written, the overflow is ignored and MIN register is set to the value written.

Set a decimal value of '00' to '59' to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is written.

MIN =0x4000_6110

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			MIN
-			0x00
-			RW

6 0	MIN	Minute Counter Register

23.2.5 HOUR: Hour counter register

HOUR is a 6-bit register that takes a value of '00' to '23' ('01' to '12' and '21' to '32' when AMPM=0, decimal) and indicates the count value of hours. It counts up when the minute counter overflows. When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the minute counter register overflows while this register is being written, the overflow is ignored and HOUR register is set to the value written.

Specify a decimal value of '00' to '23' ('01' to '12' and '21' to '32' when AMPM=0) by using BCD code according to the time system specified using bit 20 (AMPM) of real-time clock control register (RTCCR).

If the AMPM bit value is changed, the values of the HOUR register and ALARMWH register change according to the specified time system. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is written. Reset signal clears this register to 12H.

HOUR =0x4000_6114

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				HOUR
-				0x12
-				RW

5 0	HOUR	Hour counter Reigster

Table 90 shown below introduces relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 90. Specified time system by AMPM

24-hour display (AMPM = 1)				**12-hour display (AMPM = 0)**			
Time	HOUR	Time	HOUR	Time	HOUR	Time	HOUR
12 a.m.	00H	12 p.m.	12H	12 a.m.	12H	12 p.m.	32H
1 a.m.	01H	1 p.m.	13H	1 a.m.	01H	1 p.m.	21H
2 a.m.	02H	2 p.m.	14H	2 a.m.	02H	2 p.m.	22H
3 a.m.	03H	3 p.m.	15H	3 a.m.	03H	3 p.m.	23H
4 a.m.	04H	4 p.m.	16H	4 a.m.	04H	4 p.m.	24H
5 a.m.	05H	5 p.m.	17H	5 a.m.	05H	5 p.m.	25H
6 a.m.	06H	6 p.m.	18H	6 a.m.	06H	6 p.m.	26H

Table 90. Specified time system by AMPM (continued)

24-hour display (AMPM = 1)				12-hour display (AMPM = 0)			
7 a.m.	07H	7 p.m.	19H	7 a.m.	07H	7 p.m.	27H
8 a.m.	08H	8 p.m.	20H	8 a.m.	08H	8 p.m.	28H
9 a.m.	09H	9 p.m.	21H	9 a.m.	09H	9 p.m.	29H
10 a.m.	10H	10 p.m.	22H	10 a.m.	10H	10 p.m.	30H
11 a.m.	11H	11 p.m.	23H	11 a.m.	11H	11 p.m.	31H

23.2.6 DAY: Day counter register

DAY is a 6-bit register that takes a value ranging from '01' to '31' (decimal) and indicates the count value of days. It counts up when the hour counter overflows. This counter counts the followings:

- '01' to '31' (January, March, May, July, August, October, December)
- '01' to '30' (April, June, September, November)
- '01' to '29' (February, leap year)
- '01' to '28' (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the hour counter register overflows while this register is being written, this overflow is ignored and DAY register is set to the value written. Set a decimal value of '01' to '31' (corresponding to the month) to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is written.

DAY =0x4000_6118

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				DAY
-				0x00
-				RW

5 0	DAY	Day Counter Register

23.2.7 WEEK: Week counter register

WEEK is a 3-bit register that takes a value ranging from '0' to '6' (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter. When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Set a decimal value of 0 to 6 to this register. If a value outside the range is written, the reset value is written.

There is no corresponding WEEK value according to DAY. Select a value from '0' to '6' that corresponds to the day of the week and use it sequentially. After reset release, set the week counter register as you selected.

DAY =0x4000_611C

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			WEEK
-			0x0
-			RW

2 0	WEEK	Week Counter Register

23.2.8 MONTH: Month counter register

MONTH is a 5-bit register that takes a value ranging from '01' to '12' (decimal) and indicates the count value of months. It counts up when the day counter overflows. When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the day counter register overflows while this register is being written, the overflow is ignored and MONTH register is set to the value written.

Set a decimal value of '01' to '12' to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is written.

SEC =0x4000_6120

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			MONTH
-			0x00
-			RW

4 0	MONTH	Month Counter Register

23.2.9 YEAR: Year counter register

YEAR is an 8-bit register that takes a value ranging from '00' to '99' (decimal) and indicates the count value of years. It counts up when the month counter register overflows. Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the MONTH register overflows while this register is being written, the overflow is ignored and YEAR register is set to the value written.

Set a decimal value of '00' to '99' to this register in BCD code. If a value that is not a BCD code is written, it is ignored.

YEAR =0x4000_6124

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			YEAR
-			0x00
-			RW

7 0	YEAR	Year Counter Reigster

23.2.10 ALARMW: Alarm register

ALARMW sets alarm time by configuring minute, hour and week of alarm.

ALARMWM =0x4000_6128

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	ALARMWW	Reserved	ALARMWH	Reserved	ALARMWM
-	0x00	-	0x00	-	0x00
-	RW	-	RW	-	RW

22 16	ALARMWW	This register is used to set day of the week for the alarm. Each of the seven bits represents a day of the week, so that duplicate days of the week can be selected.
13 8	ALARMWH	This register is used to set hours of the alarm. Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register according to AMPM in BCD code. If a value outside the range is set, the alarm is not detected.
6 0	ALARMWM	This register is used to set minutes of the alarm. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

23.2.11 SUBDR: Sub-second counter compare register

SUBDR is a 32-bit register. This register determines the sub-second counter matching value and provides the desired frequency clock to the second counter register.

ALARMWH =0x4000_612C

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		SUBDR	
-		0x1FFFF	
-		RW	

16 0	SUBDR	Sub-second counter Compare Reigster

23.2.12 TIME_TS: TimeStamp time register

TIME_TS automatically saves the current time when a timestamp event occurs.

ALARMWW =0x4000_6130

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	HOUR_TS	MIN_TS	SEC_TS
-	0x00	0x00	0x00
-	R	R	R

23 16	HOUR_TS	HOUR stored when the timestamp event occurred
15 8	MIN_TS	MIN stored when the timestamp event occurred
7 0	SEC_TS	SEC stored when the timestamp event occurred

23.2.13 DATE_TS: TimeStamp date register

DATE_TS automatically saves the current date when a timestamp event occurs.

ALARMWW =0x4000_6134

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
YEAR_TS	MONTH_TS	WEEK_TS	DAY_TS
0x00	0x00	0x00	0x00
R	R	R	R

31 34	YEAR_TS	YEAR stored when the timestamp event occurred
23 16	MONTH_TS	MONTH stored when the timestamp event occurred
15 8	WEEK_TS	WEEK stored when the timestamp event occurred
7 0	DAY_TS	DAY stored when the timestamp event occurred

23.2.14 BKUPDR: Backup register 1 to 4

BKUPDR is a 32-bit register. This register is used to store the necessary data in the backup power mode.

BKUR1 =0x4000_6138, BKUR2 =0x4000_613C, BKUR3 =0x4000_6140, BKUR4 =0x4000_6144

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
BKUPDR			
0x00000000			
RW			

31 0	BKUPDR	Backup Register

23.2.15 RTCPFCR: RTC pin function configuration register

RTCPFCR defines function and status of the PC13, PC14 and PC15. Controls the status of PC13, PC14, and PC15 ports in all modes (RUN, IDLE, STOP, STANDBY, BACKUP POWER). When set to OUTPUT, the value of the PCU (Port Control Unit) register is ignored. When set to INPUT (Floating), the PCU.MODE register must also be set to Input.

BKUR1 =0x4000_6138, BKUR2 =0x4000_613C, BKUR3 =0x4000_6140, BKUR4 =0x4000_6144

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			PC13WKEN	Reserved	PC15MODE	PC15VALUE	PC14MODE	PC14VALUE	PC13MODE	PC13VALUE	PC13WKUEN
-			0	-	0	0	0	0	00	0	0
-			RW	-	RW	RW	RW	RW	RW	RW	RW

10 9	WKUPEN	PC13 wakeup event enable
		XX Disable wakeup event
		11 Enable wakeup event
7	PC15MODE	PC15 mode
		0 Floating.
		1 Push-pull output if LSE is disabled
6	PC15VALUE	PC15 value If the LSE is disabled and PC15MODE=1, PC15VALUE configures the PC15 output data
5	PC14MODE	PC14 mode
		0 Floating.
		1 Push-pull output if LSE is disabled
4	PC14VALUE	PC14 value If the LSE is disabled and PC14MODE=1, PC14VALUE configures the PC14 output data
3 2	PC13MODE	PC13 mode
		000 Floating.
		01 RTC_TS input **NOTE**: If timestamp wakeup is used in standby mode, PC13 must be set to timestamp function (AF7) in run mode.
		10 Push-pull output
		11 RTC_OUT push-pull output
1	PC13VALUE	PC13 value If PC13MODE=10, PC13VALUE configures the PC13 output data
0	PC13WKUPEN	Wakeup pin enable
		0 PC13 is not used as wakeup pin.
		1 PC13 is used as wakeup pin. (Only Rising Edge wakeup)
		NOTE: When this bit is set to 1, PC13 enters the input state, and PC13MODE setting is ignored. RTC time stamp operation is possible if it is enabled. PC13 wakeup event is enabled when PC13WKUPEN=1 and WKUPEN=1.
NOTE: When PC13 is used as Standby mode wakeup Pin, RTCPFCR must be initialized		

23.3 Functional description

RTC clock (fRTC) input to the sub-second counter is determined by MCCR5 and PPCLKSR.5. By setting the SUBDR register value corresponding to the RTC clock, the sub-second counter output, that is, input clock of the SEC, can be 1Hz.

In Standby mode or backup power mode, only LSE and LSI40kHz operate, so the clock source must be changed.

Table 91. SUBDR and RTC Clock Combination for 1Hz

RTCCSEL	RTCDIV	fRTC	SUBDR	ERROR
LSI (40KHz)	1	40KHz	0x0000_9C40 - 1	—
LSE (32.768KHz)	1	32.768KHz	0x0000_8000 - 1	—

23.3.1 Starting the RTC Operation

To start the RTC, a user should follow the procedure below:

1. Set the system configure register to enable RTC peripherals (PER2.29, PCER2.29, PRER2.29).
2. Select the RTC clock source by configuring MCCR5 and PPCLKSR.
3. Set the RTCEN bit to 1, while the oscillation of the clock is stable.
4. Set the necessary control registers such as alarm, timestamp, and constant-period matching etc. before activating the RTC counter.
5. Set the AMPM register for specified time system. If error correction is required, set the ERRCOR register.
6. After all the settings are completed, when the RTC counter is read for run, activate RTC by setting RTCE to 1.
7. Wait at least a clock of RTC after setting the counter when RTCEN = 1 and RTCE = 0.

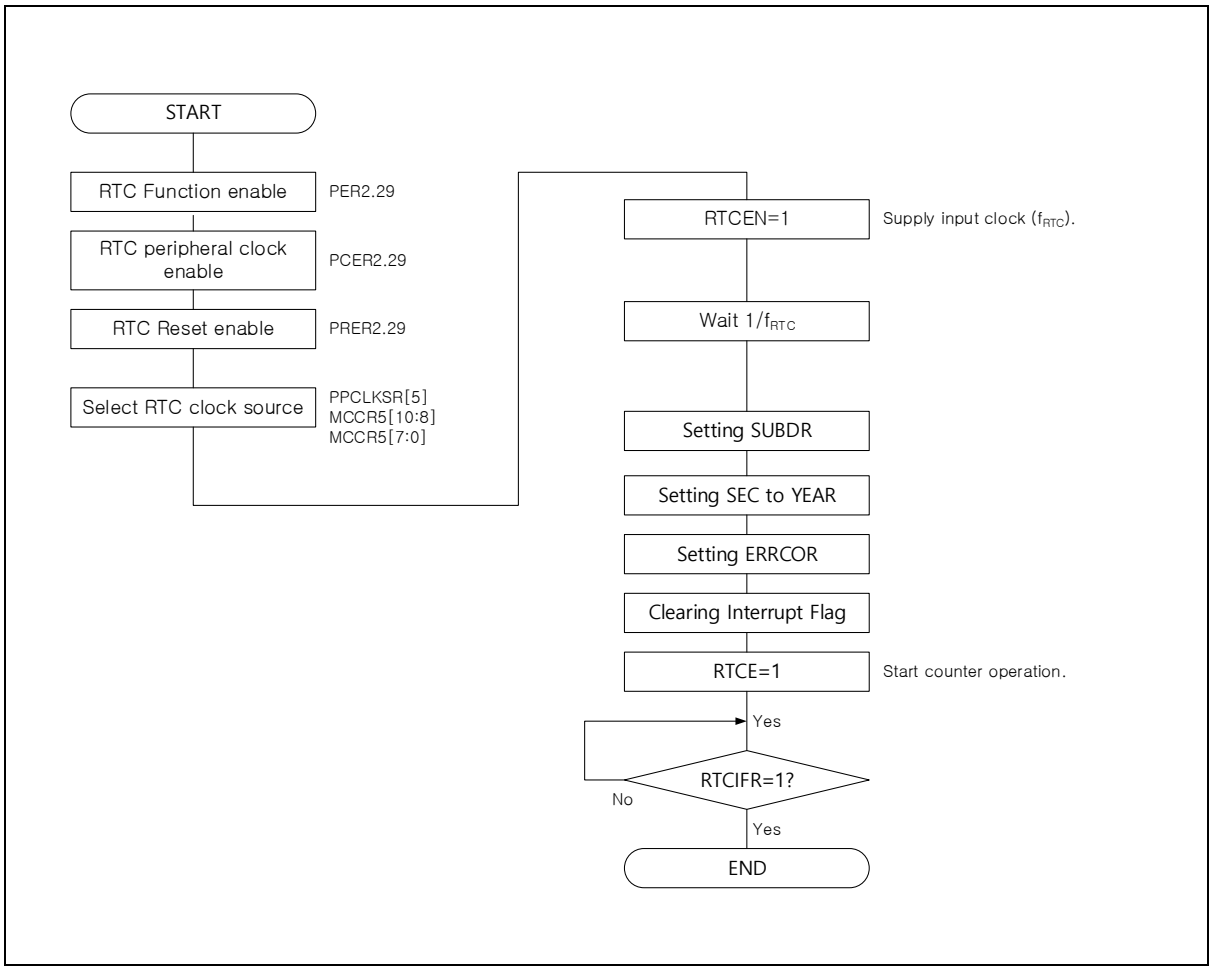


Figure 163. Start Operation Sequence of RTC

23.3.2 RTC RW mode (read and write mode)

RTC counter read and write operations are not allowed in the RTC run state. Only RTC counters can be read and written in RW mode only. The RWAIT bit must be set to 1 to enter RW mode. RWST indicates that it is in RW mode and synchronized to the RTC clock. In order to perform read or write operation, be sure to check RWST = '1' before proceeding. To exit RW mode, set RWAIT to '0'.

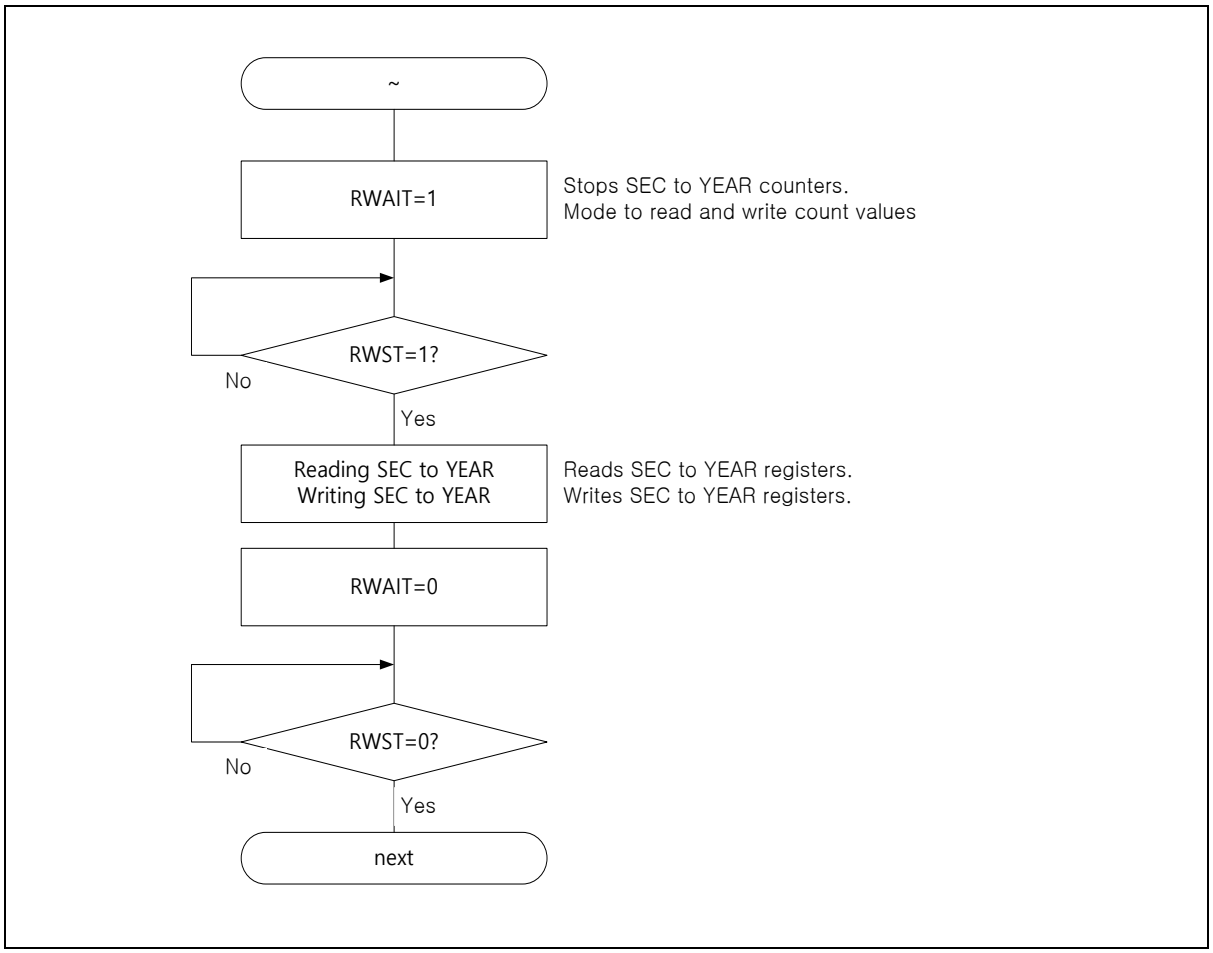


Figure 164. Read Operation Sequence of RTC

The value written in RW mode is loaded into the RTC counter at the end of RW mode. In RW mode, the SEC to YEAR counters stop counting and only the sub-second counter is continuing to count.

Therefore, even if the sub-second counter match is repeated several times in RW mode, it will be applied only once at the end of RW mode. For this reason, RW mode operation should be done in 1 second. When writing to the SEC in RW mode, even if a sub-second counter overflow occurs, it is ignored.

NOTES:

1. Complete the series of process of setting the RWAIT bit to '1' to clearing the RWAIT bit to '0' within '1' second.
2. The second counter register (SEC), minute counter register (MIN), hour counter register (HOUR), week counter register (WEEK), day counter register (DAY), month counter register (MONTH), and year counter register (YEAR) may be read/written in any sequence.
3. All registers do not have to be read and only some registers may be read or written.
4. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

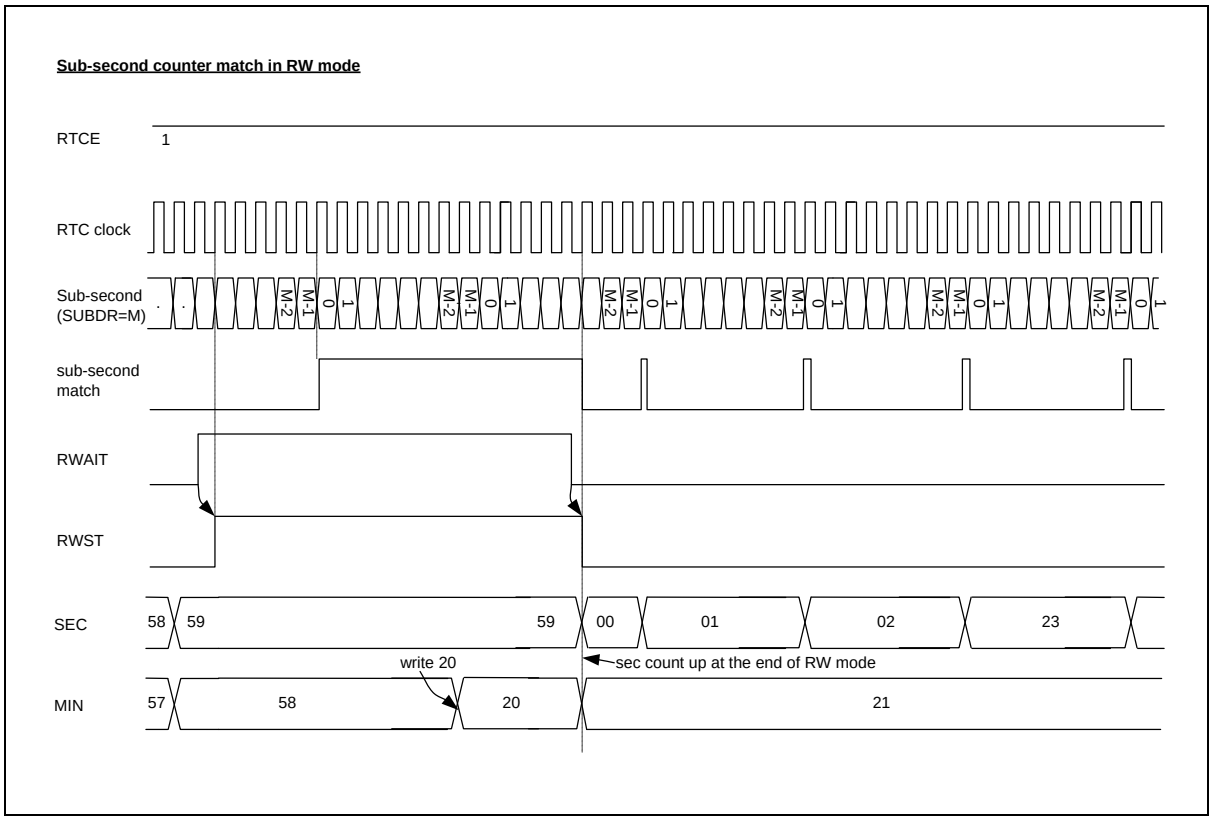


Figure 165. Sub-second Counter Match in RW Mode

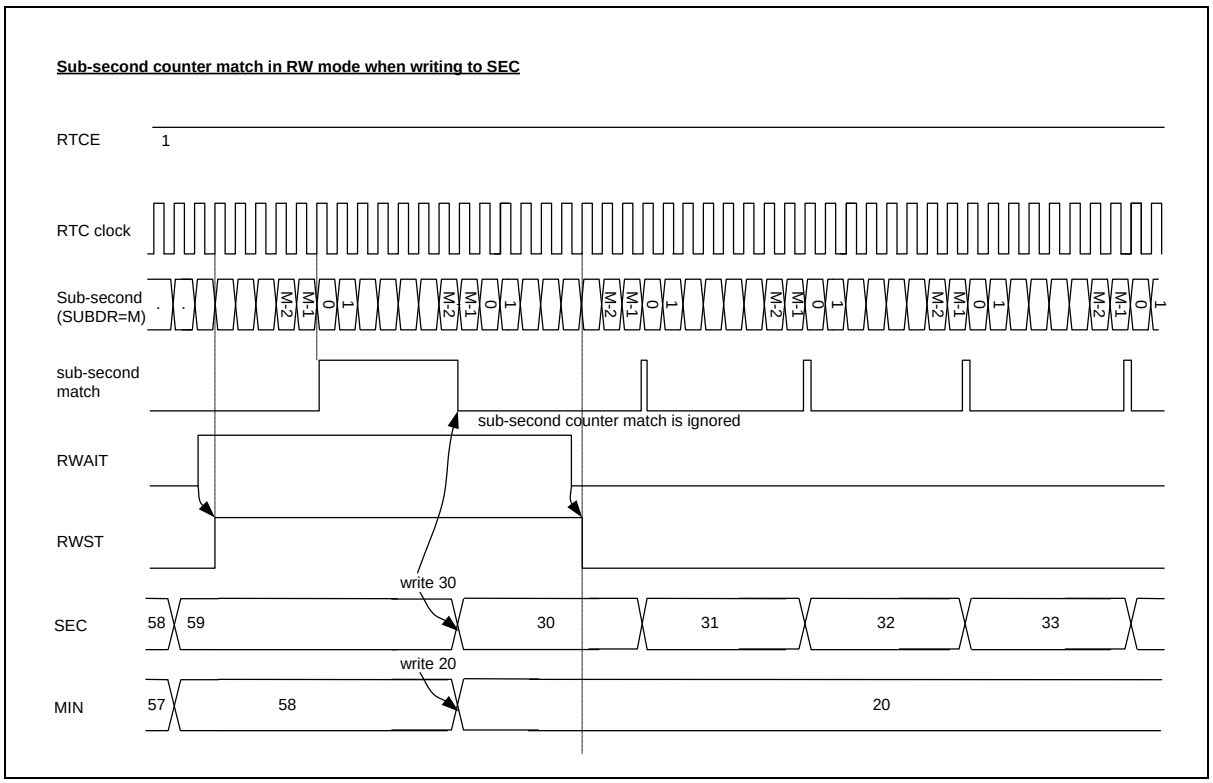


Figure 166. Sub-second Counter Match in RW Mode when Writing to SEC

23.3.3 RTC alarm setting

To activate the alarm function of RTC, write '1' to WALIE of RTCCR. Alarm time can be set with three registers ALARMWM, ALARMWH and ALARMWW. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

Check the WAFG of RTCCR to see that an alarm has occurred. Constant-period matching interrupts, alarm match interrupts and timestamp interrupts use the same interrupt source. When using these three types of interrupts at the same time, which interrupt occurred can be judged by checking the Constant-period matching flag (RIFG), the alarm detection status flag (WAFG), and the timestamp event flag (TSFG) upon RTC interrupt occurrence.

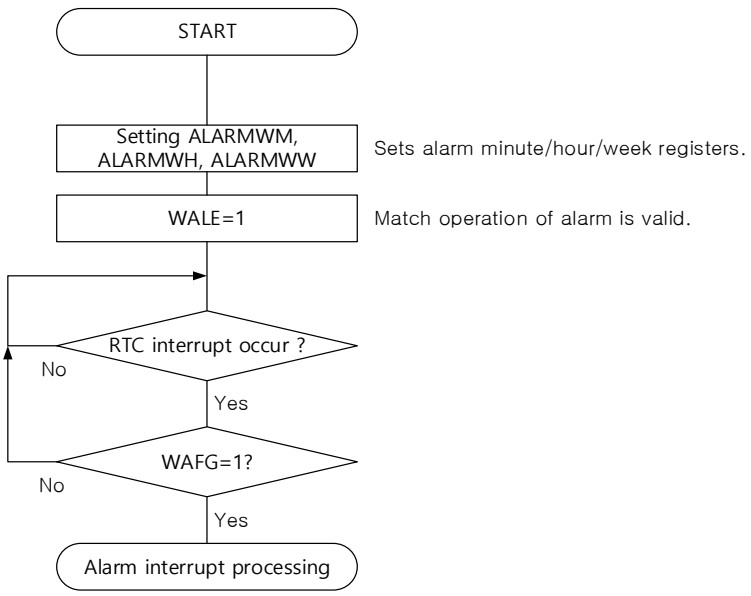


Figure 167. Alarm Operation Sequence of RTC

There is an ALARMWW bit corresponding to the Week counter register, and each bit of ALARMWW indicates the day of the week. Therefore, it is possible to set an alarm for a duplicate day of the week. Refer to Table 92 and Table 93.

Table 92. WEEK and ALARMWW Registers

WEEK	ALARMWW						
	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0
2	0	0	0	0	1	0	0

Table 92. WEEK and ALARMWW Registers (continued)

WEEK	ALARMWW						
	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3	0	0	0	1	0	0	0
4	0	0	1	0	0	0	0
5	0	1	0	0	0	0	0
6	1	0	0	0	0	0	0

Table 93. Example of Setting the Alarm

Time of alarm	Day (ALARMWW)							12-hour display		24-hour display	
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour ALARMWH	Minute ALARMWM	Hour ALARMWH	Minute ALARMWM
Every day, 0:00 a.m.	1	1	1	1	1	1	1	12	00	00	00
Every day, 1:30 a.m.	1	1	1	1	1	1	1	01	30	01	30
Every day, 11:59 a.m.	1	1	1	1	1	1	1	11	59	11	59
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	32	00	12	00
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	21	30	13	30
Monday, Wednesday, 11:59 p.m.	0	1	0	1	0	0	0	31	59	23	59

23.3.4 RTC Constant-period matching

You can cause the flag (RIFG) to occur at regular intervals. Depending on the CT value of RTCCR, a flag is generated once every 0.5 second, 1 second, 1 minute, 1 hour, 1 day, 1month. However, this period is the value when the sub-second counter outputs 1 Hz. Constant-period matching does not occur when the RTC value is zero. If the RTC interrupt is enabled on the NVIC, the Constant-period matching flag is used as the interrupt source. In this case, you can wakeup periodically in standby mode. Since the RTC interrupt and RTC alarm and RTC time stamp are also assigned together with the constant-period matching, if the RTC interrupt occurs, the interrupt handler can check the FLAG to see which interrupt has been occurred.

23.3.5 RTC Timestamp

Timestamp function is provided when external pin input occurs. Timestamp can be activated by writing 1 to TSEN in the RTCCR register, and TS_EDGE_SEL determines the edge of the timestamp input. When the Timestamp input is received, the value of the RTC counter register at that time is stored in TIME_TS, DATA_TS, and the RTC continues counting.

23.3.6 RTC 1Hz output

RTC supports 1Hz clock output to RTC_OUT pin. PC13MODE of RTCPFCR[3:2] must be set to RTCOUT. The 1Hz clock is the frequency when using LSE 32.768KHz as RTC clock and SUBDR is written 0x00007FFF. The RTC 1 Hz output can be used for error correction of the RTC counter.

23.3.7 Error correction of RTC

The watch can be corrected with high accuracy when it is slow or fast, by configuring a value to the error correction register (ERRCOR). The correction value is 1, 2, 3, 4, … $2^{\wedge 32}$-3, $2^{\wedge 32}$-2, $2^{\wedge 32}$-1 or -1, -2, -3, -4, … -$2^{\wedge 32}$-3, -$2^{\wedge 32}$-2, -$2^{\wedge 32}$-1.

Target frequency is resulted after performing the correction. The range of value that can be corrected by using the watch error correction register (ERRCOR) is shown below in Table 94 and Table 95.

Table 94. Correction Resolution by Frequency

	f_{RTC} = 32,768Hz		**f_{RTC} = 48MHz**		**f_{RTC} = 40,000Hz**	
Correction Data	DEV = 0	DEV = 1	DEV = 0	DEV = 1	DEV = 0	DEV = 1
Minimum resolution	±1.53ppm	±0.51ppm	±0.001ppm	±0.0003ppm	±1.25ppm	±0.41ppm
1	±1.53 ppm	±0.51 ppm	±0.0010 ppm	±0.0003 ppm	±1.25 ppm	±0.41 ppm
2	±3.05 ppm	±1.02 ppm	±0.0021 ppm	±0.0007 ppm	±2.50ppm	±0.83 ppm
3	±4.58 ppm	±1.52 ppm	±0.0031 ppm	±0.0010 ppm	±3.75ppm	±1.25 ppm
122	±186.2 ppm	±62.1 ppm	±0.1271 ppm	±0.0423 ppm	±152.5 ppm	±50.8 ppm
123	±187.7 ppm	±62.6 ppm	±0.1281 ppm	±0.0427 ppm	±153.7 ppm	±51.2ppm
124	±189.2 ppm	±63.0 ppm	±0.1292 ppm	±0.0431 ppm	±155.0 ppm	±51.7 ppm
1024	±1562.5 ppm	±520 ppm	±1.0667 ppm	±0.3556 ppm	±1280.0 ppm	±427 ppm
32768	±50000 ppm	±16667 ppm	±34.133 ppm	±11.378 ppm	±40960 ppm	±13653 ppm
1048576	—	—	±1092.27 ppm	±364.09 ppm	—	—
28800000	—		±30000 ppm	±10000 ppm	—	—

Example of calculating the correction value

Following expression is used for calculation of the correction value which is used when correcting the count value of internal counter (32-bit):

- (DEV = 0): Correction value = Number of correction counts in 1 minute ÷ 3 = (fRTC ÷ 32768 - 1) x (32768 x 60 ÷ 3)
- (DEV = 1): Correction value = Number of correction counts in 1 minute = (fRTC ÷ 32768 - 1) x (32768 x 60)

Correction example

Example procedure to correct from 32772.3Hz to 32768Hz (32772.3 Hz – 131.2 ppm) is shown below:

1. Measuring the oscillation frequency: The fRTC is measured by outputting to RTC_OUT pin when the ERRCOR register is set to its initial value (0x00000000).
2. Calculating the correction value: Assume the target frequency to be 32768 Hz (32772.3Hz–131.2ppm) and DEV to be 0.
   - Correction value = (fRTC ÷ 32768 – 1) × 32768 × 60 ÷ 3 = (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3 = 86
3. Calculating the values to be set to (ERRCOR): If the correction value is 0 or larger (when slowing), assume DIR to be 1.

   Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the error correction register such that DEV is 0 and the correction value is 86 results in 32768 Hz (0 ppm)

   - ERRCOR = 0x5000_0056

NOTE: If the CORDATA value exceeds the clock frequency range, it may malfunction.

23.3.8 BKUPDR and RTCPFCR registers

BKUPDR Register use only backup power mode. This register is retained when wakeup in backup power mode. In Backup power mode, it is wakeup after storing the value in BKUPDR register. If BKUPDR register value is held in Run state, it can be confirmed that wakeup is normally performed in backup power mode. For example can change the Timer Duty. See the picture below.

When wakeup in backup power mode, the timer register is initialized and must be set again.

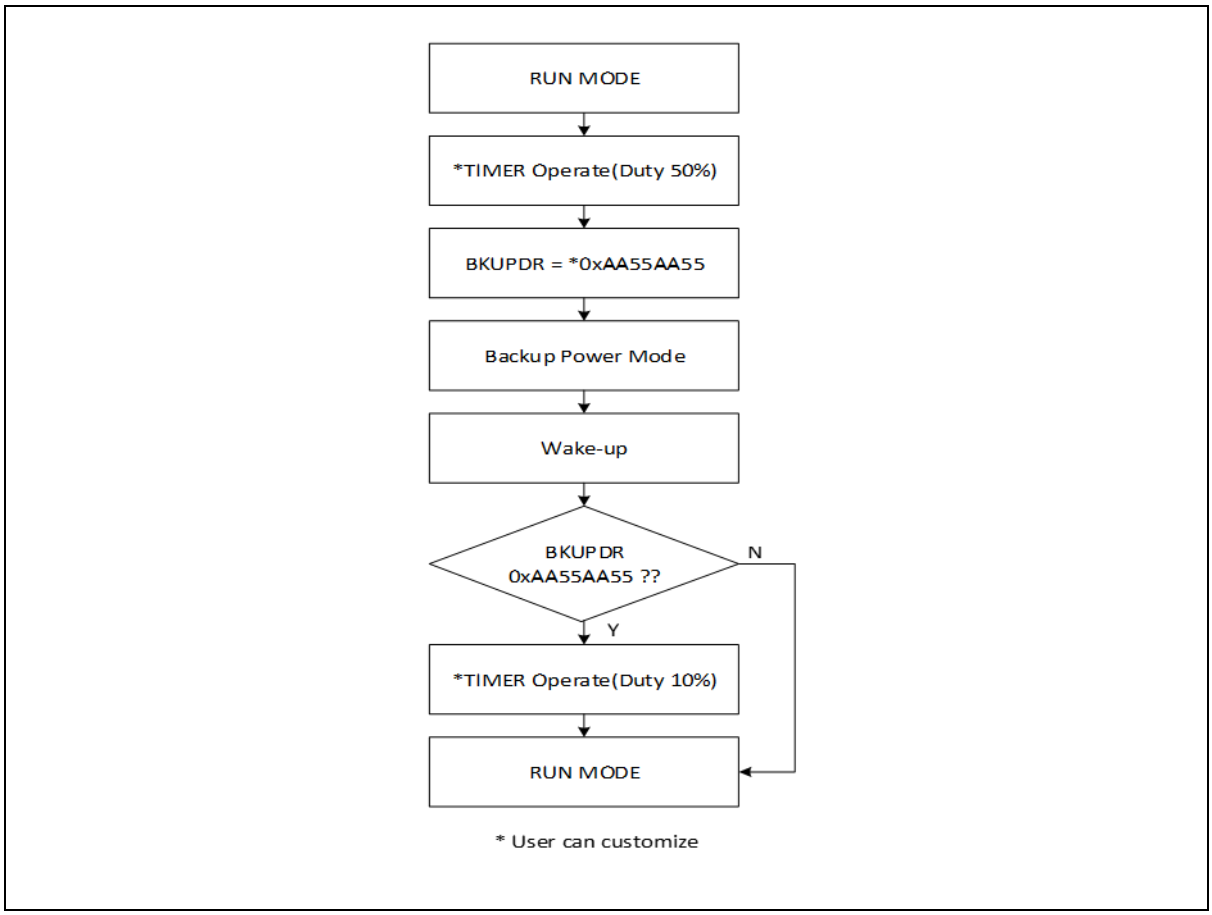


Figure 168. Configuring Sequence of BKUPDR Register

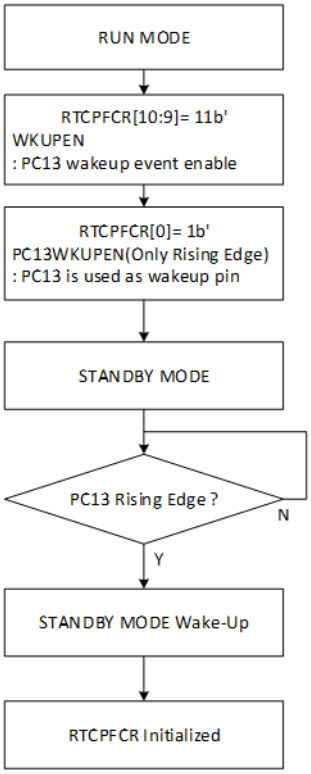
In Standby & Backup power mode, GPIO ports except PC13, PC14 and PC15 cannot be used. Therefore, it is used to check RTC output (PC13) and PC14 and PC15 pins operating in standby & backup power mode, and Wakeup (PC13) in STANDBY mode.

PC14, PC15 cannot be used as push pull output when LSE is used with SXIN SXOUT pin of LSE. Must not use LSE to use PC14, PC15 as PUSH PULL OUTPUT.

Table 95. PC13, PC14, PC15 Function in RTCPFCR register

	Floating	Push-Pull output	RTC_TS	RTC_OUT	*Standby Wake-up Source
PC13	O	O	O	O	O
PC14	O	O	X	X	X
PC15	O	O	X	X	X

PC13 pin wakeup of the RTCPFCR is used only in Standby mode and is not related to the wakeup of the RTC block. Figure 169 shows sequence of PC13 pin wakeup operation.



NOTE:

1. PC13 Pin wake-up only works with Rising Edge, and RTCPFCR must be initialized after wakeup.

Figure 169. PC13 wakeup in Standby Mode

24. Temp sensor

Temp sensor is to use the internal oscillator LSITS by default, which has a large temperature variation. The temperature-dependent LSITS frequency can be calculated based on a precisely trimmed internal oscillator or an external clock. Reference clock and sense clock of the temp sensor can be changed by configuring TSENSECON register. When selecting the clock, frequency of the reference clock must be faster than the sense clock frequency. In SCU, each clock must be activated by configuring corresponding register. If value of TSREFCNT using REF clock matches the TSREFPERIOD set by the user, a match flag is generated and frequency of the sense clock is calculated by reading the value of TSSENSECNT at this time. Match flags can be used as interrupt sources.

Note) In order to use the sensor properly, it should be set **the reference clock of temperature sensor is minimum 8MHz.**

Glossary for this chapter

- HSI_I: HSI clock set by CSCR
- MCLK: System clock set by SCCR
- LSE: External sub oscillator
- LSITS: Internal temp sense oscillator
- LSI750KHz: Internal 750KHz oscillator
- LSI40KHz: Internal 40KHz oscillator

24.1 Temp sensor block diagram

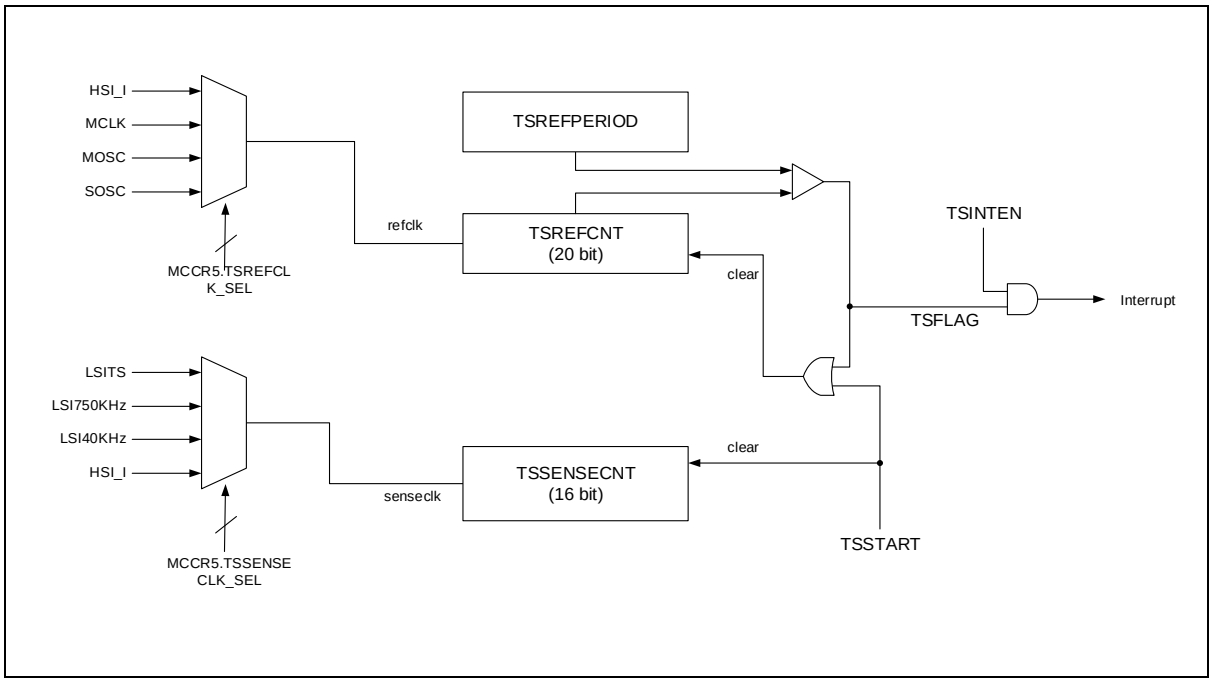


Figure 170. Temp Sensor Block Diagram

24.2 Registers

Base address of a temp sensor block is introduced in the followings:

Table 96. Base Address of TSENSE Interface

Name	Base address
TSENSE	0x4000_6300

Table 97. TSENSE Register Map

Name	Offset	Type	Description	Reset value	Reference
TSENSECON	0x0000	RW	Temp Sensor Control Register	0x0000_0000	24.2.1
TSREFPERIOD	0x0004	RW	Temp Sensor Reference Clock period Register	0x000F_FFFF	24.2.2
TSSENSECNT	0x0008	R	Temp Sensor Sense Clock Count Register	0x0000_0000	24.2.3

24.2.1 TSENSECON: Temp sensor control register

TSENSECON =0x4000_6300

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				TSSINTEN	TSSFLAG	TSSTART	TSSEN
-				0	0	0	0
-				RW	RW	RW	RW

3	TSSINTEN	Temp Sensor interrupt enable	
		0	Temp sensor interrupt disabled
		1	Temp sensor interrupt enabled
2	TSSFLAG	Temp Sensor CNT match flag	
		0	Temp sensor CNT match flag is not generated
		1	Temp sensor CNT match flag not generated
		NOTE: This flag is reset when "1" is written to it. Writing "0" to it is invalid.	
1	TSSTART	Temp Sensor start	
		0	Wait
		1	Clear & start
0	TSSEN	Temp Sensor enable	
		0	Temp Sensor disabled
		1	Temp Sensor enabled

24.2.2 TSREFPEROID: Temp sensor reference clock period register

TSENSECON =0x4000_6304

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		TSREFPERIOD	
-		0xFFFFF	
-		RW	

19 0	TSREFPERIOD	Temp Sensor Reference Clock Period Register

24.2.3 TSSENSECNT: Temp sensor sense clock count register

TSENSECON =0x4000_6308

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		TSSENSECNT	
-		0x00	
-		R	

15 0	Temp Sensor Reference Count Register

24.3 Functional discription

24.3.1 How to calculate and convert to temperature

To convert to temperature, two temperature values in the region of OTP should be read.

The region of OTP includes the specific value of two temperature(85°C and 30°C). These value calculate to slope and offset of temperature. There is a guide related to converting to temperature. Refer to the Table 98

Table 98. Specific temperature value in the region of OTP

Variable	Temperature	Address of OTP	Note
Specific value	85°C	0x3F0103F4	Value characteristics vary by sample
	30°C	0x3F0103F8	

Convert the temperature as shown below.

A. Read two specific values about temperature(85°C and 30°C)

B. Divide each value by 8192(0x2000) and multiply system clock / TSREFPEROID register.

C. 85°C is X, 30°C is Y. (8192 is a fixed value)

D. Slope = (85 – 30) / (X – Y) , Offset = 30 – (Slope x Y)

E. Temperature(°C) = Slope x TSSENSECNT + offset

For example, Reference period of temperature is 1kHz. TSREFPEROID is system clock / 1000.

25. Electrical characteristics

25.1 Absolute maximum ratings

Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions.

Table 99. Absolute maximum rating

Parameter	Symbol	Ratings	Unit	Note
Supply Voltage	VDD	-0.3 – +6.5	V	—
Normal Pin	V_I	-0.3 – VDD+0.3	V	Voltage on any pin with respect to VSS
	V_O	-0.3 – VDD+0.3	V	
	I_{OH}	25	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	ΣI_{OH}	100	mA	Maximum current (ΣI_{OH})
	I_{OL}	22	mA	Maximum current sunk by (I_{OL} per I/O pin)
	ΣI_{OL}	88	mA	Maximum current (ΣI_{OL})
Total Power Dissipation	T_P	300	mW	—
Storage Temperature	T_{STG}	-55 – +125	℃	—

25.2 Recommended operating conditions

Table 100. Recommended Operating Condition

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply Voltage	VDD	—	1.8	—	5.5	V
		Use USB function	3.0	—	5.5	V
Operating Frequency	FREQ	HSE	2	—	16	MHz
		LSE	—	32.768	—	KHz
		HSI	46.56	48	49.44	MHz
		LSI750kHz	600	750	900	KHz
		LSI40kHz	22.8	40	62.6	KHz
Operating Temperature	Top	Top	-40	—	+85	℃
Supply Rise Rate	t_{rVDD}	—	—	—	10	V/ms
Supply Fall Rate	t_{fVDD}	—	—	—	10	V/ms

NOTE: AVDD must always be equal to or greater than VDDEXT

25.3 ADC characteristics

Table 101. ADC Electrical Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating voltage	AVDD		2.7	5	5.5	V
Resolution					12	Bit
Operating current	IDDA	AVDD = 5.0V @f_{MCLK} = 24MHz		1.6		mA
Analog input range	V_{AN}		VSS		AVDD	V
Conversion rate	F_{CONV}	@AVDD > 3.6V		—	1.5	MHz
		@AVDD > 3.0V			1	MHz
		@AVDD > 2.7V			0.5	MHz
Operating frequency	ACLK				25	MHz
DC accuracy	INL	AVDD = 5.0V		±3	±6	LSB
	DNL	AVDD = 5.0V		±2	±3	LSB
Error Of Bottom	EOB			±4		LSB
Error Of Top	EOT			±4		LSB

NOTE) AVDD must always be equal to or greater than VDDEXT

25.4 DAC characteristics

Table 102. DAC Electrical Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating voltage	AVDD		2.7	5	5.5	V
Resolution	-	-	-	10		BIT
Load Resistance with Buffer	R_{LOAD}	Load connected to VSS	5	-	-	kΩ
		Load connected to VAVDD	25	-	-	kΩ
Output Impedance Without Buffer	R_O		-	-	130	kΩ
DAC Output Voltage	D_{AOUT}		0.2	-	AVDD - 0.2	V
Operating Current	$I_{AVDD,rms}$	No load, middle code(0x200)	-	1.48	0.95	mA
DNL	-	No R-Load, DAC Output	-	±1	±2	LSB
	-	R-Load=5kΩ, C-Load=50pF, DAC Output	-	±2	±8	
INL	-	No R-Load, DAC Output	-	±3	±5	LSB
	-	R-Load=5kΩ, C-Load=50pF, DAC Output	-	±4	±8	
Offset	-	Offset Error is difference between measured value at Code (0x200) and the ideal value(AVDD/2)	-	-	8	LSB
Gain Error	-		-	1.0	-	%
Conversion Time	$t_{SETTLINB}$		-	2	4	us
Stop Current			-	2	400	nA

NOTE

1. AVDD must always be equal to or greater than VDDEXT
2. Data based on characterization results, not tested in production

25.5 Power on reset characteristics

Table 103. POR Electrical Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
POR set level	V_{set}	—	1.05	1.2	1.35	V
POR reset level	V_{reset}	—	0.9	1.1	1.3	V

25.6 Low voltage reset/indicator characteristics

Table 104. Low Voltage Reset/Indicator Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol		Conditions	Min	Typ	Max	Units
Operating voltage	VDD			1.5	5	5.5	V
Detection level	V_{LVR}	Level0	Rising voltage	1.56	1.68	1.80	V
			Falling voltage	-	1.63	1.74	
		Level1	Rising voltage	1.65	1.77	1.89	
			Falling voltage	1.60	1.72	1.84	
		Level2	Rising voltage	1.74	1.87	2.00	
			Falling voltage	1.70	1.82	1.94	
		Level3	Rising voltage	1.85	1.99	2.13	
			Falling voltage	1.81	1.94	2.07	
		Level4	Rising voltage	1.95	2.09	2.23	
			Falling voltage	1.89	2.03	2.17	
		Level5	Rising voltage	2.07	2.22	2.37	
			Falling voltage	2.01	2.16	2.31	
		Level6	Rising voltage	2.23	2.40	2.57	
			Falling voltage	2.19	2.35	2.51	
		Level7	Rising voltage	2.41	2.59	2.77	
			Falling voltage	2.35	2.52	2.69	
		Level8	Rising voltage	2.61	2.80	2.99	
			Falling voltage	2.53	2.72	2.91	
		Level9	Rising voltage	2.96	3.18	3.40	
			Falling voltage	2.89	3.10	3.31	
		Level10	Rising voltage	3.10	3.33	3.56	
			Falling voltage	3.02	3.24	3.46	
		Level11	Rising voltage	3.50	3.76	4.02	
			Falling voltage	3.41	3.66	3.91	
		Level12	Rising voltage	3.63	3.90	4.17	
			Falling voltage	3.53	3.79	4.05	
		Level13	Rising voltage	3.93	4.22	4.51	
			Falling voltage	3.83	4.11	4.39	
		Level14	Rising voltage	4.09	4.40	4.71	
			Falling voltage	3.99	4.28	4.57	
		Level15	Rising voltage	4.37	4.69	5.01	
			Falling voltage	4.26	4.57	4.88	
Hysteresis	—			—	100	200	mV
Noise cancelling time	—			—	2	-	us
Operation current	I_{DD}			—	3.5	5	uA
Operation current(STOP)	$I_{DD, STOP}$			—	2.5	3	nA

25.7 High frequency internal RC oscillator characteristics

Table 105. High Frequency Internal RC Oscillator Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating current	I_{HIRC}	Enable	—	350	450	uA
		Disable	—	0.6	70	nA
Frequency	f_{HIRC}	VDD = 1.8V to 5.5V T_A = – 40°C to +85℃	46.56	48	49.44	MHz
		VDD = 1.8V to 5.5V T_A = 0°C to +50℃	47.28	48	48.72	MHz

25.8 Low frequency internal RC oscillator characteristics

Table 106. Low Frequency (750KHz) Internal RC Oscillator Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		1.8	5	5.5	V
Operating current	I_{LIRC}	Enable	—	1.5	2	uA
		Disable	—	1	20	nA
Frequency	f_{LIRC}	VDD = 1.8V to 5.5V	600	750	900	kHz
Stabilization time	t_{LFS}	—	—	100		us

Table 107. Low Frequency (40KHz) Internal RC Oscillator Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		1.8	5	5.5	V
Operating current	I_{LIRC}	Enable	—	305	588.2	nA
		Disable	—	—	1	nA
Frequency	f_{LIRC}	VDD = 1.8V to 5.5V	22.8	40	62.6	KHz
Stabilization time	t_{LFS}	—	—	100		us

25.9 DC electrical characteristics

Table 108. DC Electrical Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input high voltage	V_{IH}	PA,PB,PC,PF, nRESET, nBOOT	0.8VDD	—	VDD	V
Input low voltage	V_{IL}	PA,PB,PC,PF, nRESET, nBOOT	—	—	0.2VDD	V
Input hysteresis	△V	All input pins, nRESET, nBOOT VDD=3V	100	200	—	mV
Output high voltage	V_{OH1}	VDD=5V, I_{OH1} = – 2.64mA; PA[15:11],PA[4:0],PB[12:6], PB[2],PC[12:4],PC[1:0], PF[2:0]	0.8VDD	—	—	V
	V_{OH2}	VDD=5V, I_{OH2} = – 4mA; PA[10:5], PB[15:13, 5:3, 1:0], PC[3:2]	0.8VDD	—	—	V
	V_{OH3}	VDD=5V, I_{OH3} = – 1mA; PC[15:13]	VDD-0.5	—	—	V
Output low voltage	V_{OL1}	VDD=5V, I_{OL1}=3.6mA; PA[15:11],PA[4:0],PB[12:6], PB[2],PC[15:4],PC[1:0],PF[2:0]	—	—	0.2VDD	V
Output low voltage	V_{OL2}	VDD=5V, I_{OL2}=3.6mA; PA[10:5], PB[15:13, 5:3, 1:0], PC[3:2]	—	—	0.2VDD	V
Input high leakage current	I_{IH}	All Input ports	—	—	1	uA
Input low leakage current	I_{IL}	All Input ports	– 1	—	—	uA
Pull-up resistor	R_{PU}	V_I=0V, T_A=25°C, VDD=5V PA,PB,PC,PF	25	50	100	KΩ
		V_I=0V, T_A=25°C, VDD=5V nRESET, nBOOT	150	250	400	
Pull-down resistor	R_{PD}	V_I=VDD, T_A=25°C, VDD=5V PA,PB,PC,PF	25	50	100	KΩ

25.10 Supply current characteristics

Table 109. Supply Current Characteristics

(Temperature: -40 to +85℃)

Parameter	Symbol	Conditions		Typ	Max	Units
Supply current	I_{DD1} (Run)	f_{XIN} = 8MHz	VDD=5V±10% All peripherals off	3	—	mA
		f_{HIRC} = 12MHz		2.5	—	
		f_{HIRC} = 48MHz		8	—	
		F_{LIRC} = 750KHz		250	—	uA
		F_{LSE} = 32.768KHz		1	—	
	I_{DD2} (Sleep)	f_{XIN} = 8MHz	VDD=5V±10% All peripherals off	2.5	—	mA
		f_{HIRC} = 12MHz		2	—	
		f_{HIRC} = 48MHz		6	—	
		F_{LIRC} = 750KHz		250	—	uA
		F_{LSE} = 32.768KHz		1	—	
	I_{DD3} (Stop)	All Oscillators Off	VDD=5V±10%	2	—	uA
		Only LSE on		3.5	—	
		Only LSI40K on		3.5	—	
		Only LVR on		6	—	
	I_{DD4} (Standby)	All Oscillators Off	VDD=5V±10%	1.5	—	uA
		Only LSE on		2	—	
		Only LSI40K on		2	—	
		Only LVR on		5	—	
	I_{DD5} (Backup power)	All Oscillators Off	VDD=5V±10%	1.3	—	uA
		Only LSE on		1.6	—	
		Only LSE + RTC on		1.7	—	
		Only LSI40K on		1.5	—	

NOTES:

1. Where the f_{XIN} is an external main oscillator, the f_{SUB} is an external sub oscillator, the f_{HIRC} is a high frequency internal RC oscillator, and the f_x is the selected system clock.
2. All supply current items don't include the current of a low frequency internal RC oscillator and a peripheral block.
3. All supply current items include the current of the power-on reset (POR) block.

25.11 AC characteristics

Table 110. AC Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
RESETB input low width	t_{RST}	VDD = 5V	10	—	—	us
Interrupt input high low width	t_{IWH}, t_{IWL}	All interrupts, VDD = 5V	100	—	—	ns
External counter input high low pulse width	t_{ECWH}, t_{ECWL}	VDD = 5V All external counter input	100	—	—	
External counter transition time	t_{REC}, t_{FEC}	Ecn, VDD = 5V All external counter input	—	—	20	

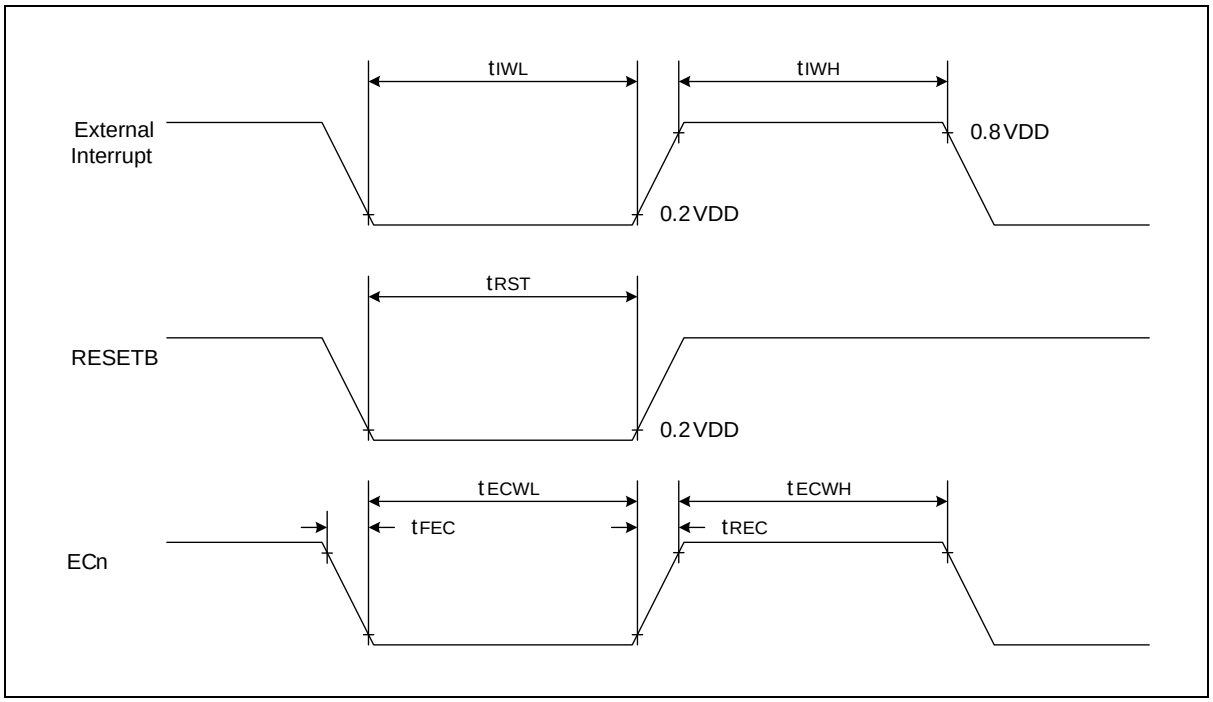


Figure 171. AC Timing

25.12 USART SPI characteristics

Table 111. SPI Characteristics with High Voltage

(Temperature: -40℃ to +85℃, VDD = 3V to 5.5V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
SPI Clock frequency	f_{SCK} $1/f_{c(SCK)}$	Master mode	—	—	12	MHz
		Slave mode	—	—	6	
Output clock pulse period	t_{SCK}	Internal SCK source	80	—	—	ns
Input clock pulse period		External SCK source	80	—	—	
Output clock high, low pulse width	t_{SCKH}, t_{SCKL}	Internal SCK source	16	—	—	
Input clock high, low pulse width		External SCK source	16	—	—	
First output clock delay time	t_{FOD}	Internal/external SCK source	40	—	—	
Output clock delay time	t_{DS}	—	—	—	25	
Input setup time	t_{DIS}	—	36`	—	—	
Input hold time	t_{DIH}	—	36	—	—	

Note) The speed of support for SPI clock frequency depends on the VDD value.

Table 112. SPI Characteristics with Low Voltage

(Temperature: -40℃ to +85℃, VDD = 1.8V to 3V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
SPI Clock frequency	f_{SCK} $1/f_{c(SCK)}$	Master mode	—	—	6	MHz
		Slave mode	—	—	6	
Output clock pulse period	t_{SCK}	Internal SCK source	160	—	—	ns
Input clock pulse period		External SCK source	160	—	—	
Output clock high, low pulse width	t_{SCKH}, t_{SCKL}	Internal SCK source	32	—	—	
Input clock high, low pulse width		External SCK source	32	—	—	
First output clock delay time	t_{FOD}	Internal/external SCK source	80	—	—	
Output clock delay time	t_{DS}	—	—	—	50	
Input setup time	t_{DIS}	—	72	—	—	
Input hold time	t_{DIH}	—	72	—	—	

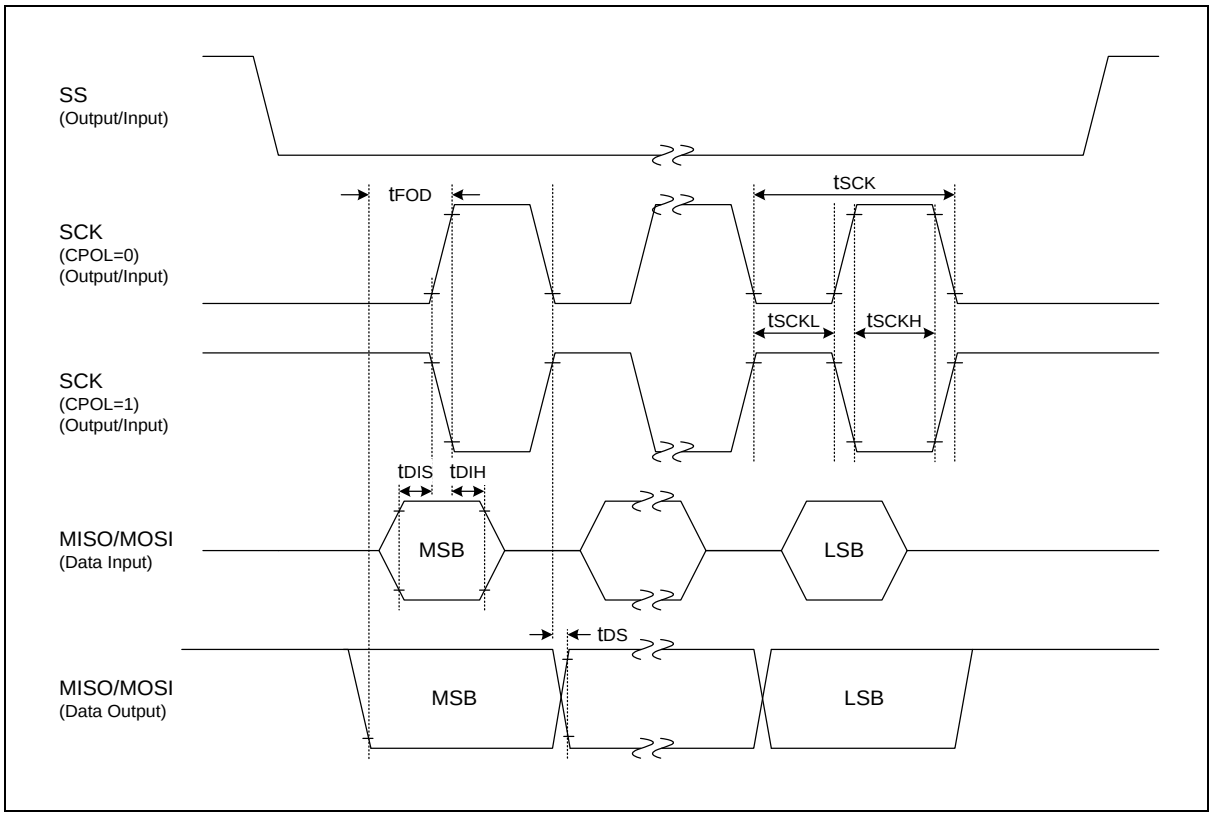


Figure 172. SPI Timing

25.13 I2C characteristics

Table 113. I2C Characteristics

(Temperature: -40℃ to +85℃, VDD = 1.8 to 5.5V)

Parameter	Symbol	Standard		Fast		Fast Plus		Units
		Min	Max	Min	Max	Min	Max	Units
Clock frequency	t_{SCL}	0	100	0	400	0	1000	KHz
Clock high pulse width	t_{SCLH}	4.0	—	0.6	—	0.26	—	us
clock low pulse width	t_{SCLL}	4.7	—	1.3	—	0.5	—	
Bus free time	t_{BF}	4.7	—	1.3	—	0.5	—	
Start condition setup time	t_{STSU}	4.7	—	0.6	—	0.26	—	
Start condition hold time	t_{STHD}	4.0	—	0.6	—	0.26	—	
Stop condition setup time	t_{SPSU}	4.0	—	0.6	—	0.26	—	
Stop condition hold time	t_{SPHD}	4.0	—	0.6	—	0.26	—	
Output valid from clock	t_{VD}	0	—	0	—	0	—	
Data input hold time	t_{DIH}	0	—	0	1.0	0	0.45	
Data input setup time	t_{DIS}	250	—	100	—	50	—	ns

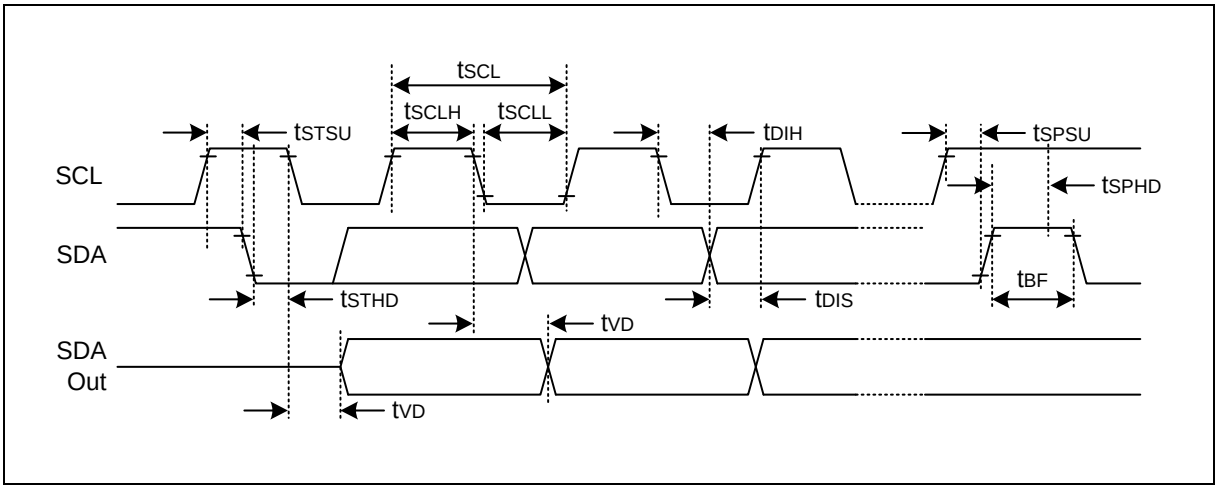


Figure 173. I2C Timing

25.14 USART UART timing characteristics

Table 114. UART Timing Characteristics

(Temperature: -40℃ to +85℃, VDD = 1.8 to 5.5V)

Parameter	Symbol	Min	Typ	Max	Units
Serial port clock cycle time	t_{SCK}	1250	t_{CPU} x 16	1650	ns
Output data setup to clock rising edge	t_{S1}	590	t_{CPU} x 13	—	
Clock rising edge to input data valid	t_{S2}	—	—	590	
Output data hold after clock rising edge	t_{H1}	t_{CPU} – 50	t_{CPU}	—	
Input data hold after clock rising edge	t_{H2}	0	—	—	
Serial port clock High, Low level width	t_{HIGH}, t_{LOW}	470	t_{CPU} x 8	970	

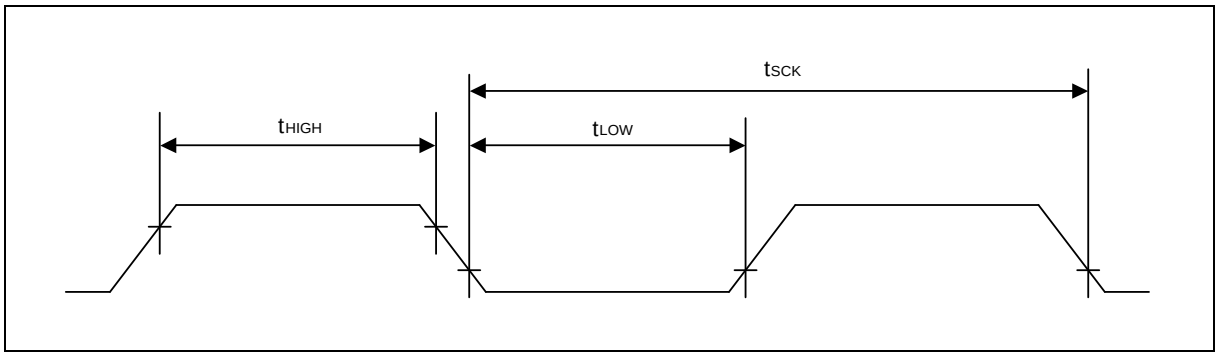


Figure 174. Waveform of UART Timing Characteristics

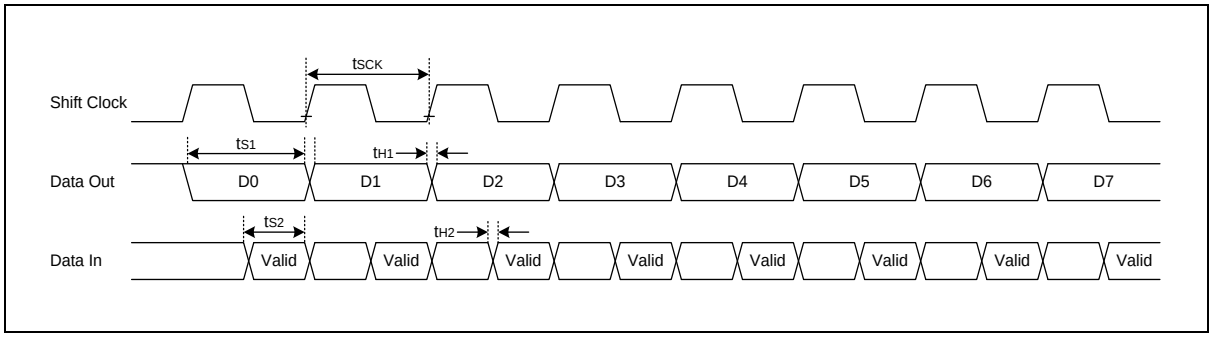


Figure 175. UART Module Timing

25.15 Data retention voltage in STOP mode

Table 115. Data Retention Voltage in STOP mode

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Data retention supply voltage	V_{DDDR}	—	1.8	—	5.5	V
Data retention supply current	I_{DDDR}	V_{DDDR} = 1.8V, (T_A = 25°C), Deep sleep mode	—	—	1	uA

25.16 Internal Flash ROM characteristics

Table 116. Internal Flash ROM Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Page write time	t_{FSW}	—	—	30	—	us
Page erase time	t_{FSE}	—	—	4	—	ms
Chip erase time	t_{FCE}	—	—	8	—	ms
Read Access Time	t_{FRA}	—	30	—	80	ns
Flash program voltage	V_{PGM}	On erase/write	1.65	—	5.5	V
Endurance of write/erase	N_{FWE}	T_A=25 ˚C, Page unit	100,000	—	—	Times
Retention time	t_{FRT}		10	—	—	Years

25.17 Main oscillator characteristics

Table 117. Main Oscillator Characteristics

(Temperature: -40℃ to +85℃)

Oscillator	Parameter	Conditions	Min	Typ	Max	Units
Operating voltage	VDD	—	1.8	5.0	5.5	V
Operating current	IDD	—	—	—	2.5	mA
Power down current	I_{STOP}	—	—	0.2	30	nA
Output frequency	XOUT input frequency	VDDEXT≥1.8V ISEL_I<1:0>=2'b11	2.0	—	4.0	MHz
		VDDEXT≥2.0V ISEL_I<1:0>=2'b10	2.0	—	8.0	MHz
		VDDEXT≥2.2V ISEL_I<1:0>=2'b01	2.0	—	12.0	MHz
		VDDEXT≥2.4V ISEL_I<1:0>=2'b00	2.0	—	16	MHz
Start-up time	T_{start}	—	—	2	—	ms
Crystal input (low)	V_{IL}	—	—	—	0.2VDD	V
Crystal input (high)	V_{IH}	—	0.8VDD	—	—	V
Crystal out (low)	V_{OL}	—	—	—	0.2VDD	V
Crystal out (high)	V_{OH}	—	0.8VDD	—	—	V
External load cap	C_L	2M<f_{OUT}<4M	18	30	35	pf
		4M<f_{OUT}<12M	10	22	30	pf
		12M<f_{OUT}<16M	7	18	22	pf
Feedback resistance	R_{FB}	VDDEXT=5V	0.7	1.0	1.3	MΩ

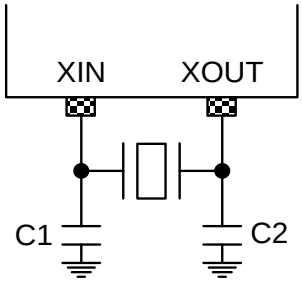


Figure 176. Crystal/Ceramic Oscillator

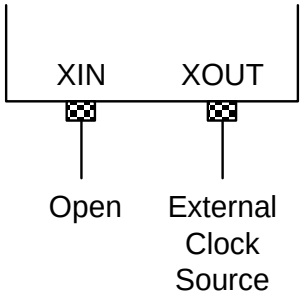


Figure 177. External Clock

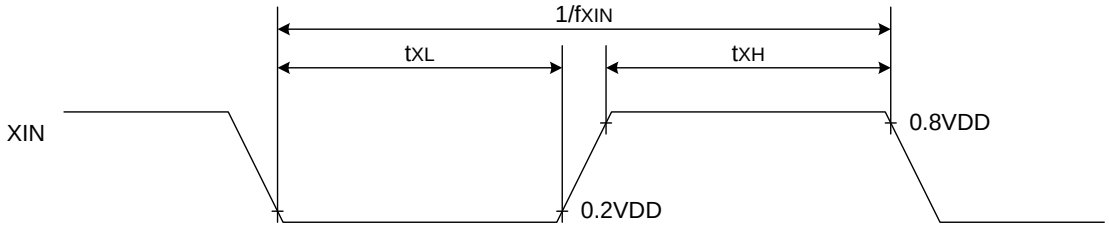


Figure 178. Clock Timing Measurement at XIN

25.18 Sub oscillator characteristics

Table 118. Sub Oscillator Characteristics

(Temperature: -40℃ to +85℃)

Oscillator	Parameter	Conditions	Min	Typ	Max	Units
Operating voltage	VDD	—	2.7	5.0	5.5	V
Operating current	IDD	ISEL_I[1:0]=2'b11 CL=6pf, Schmitt Off	—	0.26	0.82	uA
		ISEL_I[1:0]=2'b10 CL=7pf, Schmitt Off		0.3	0.94	uA
		ISEL_I[1:0]=2'b01 CL=7pf, Schmitt Off		0.34	1.13	uA
		ISEL_I[1:0]=2'b00 CL=12.5pf, Schmitt Off		0.49	1.44	uA
Power down current	I_{STOP}	—	—	—	15	nA
Output frequency	f_{SUB}	—	—	32.768	—	kHz
Start-up time	T_{start}	—	—	2	—	s
Crystal input (low)	V_{IL}	—	—	—	0.2VDD	V
Crystal input (High)	V_{IH}	—	0.8VDD	—	—	V
Crystal out (low)	V_{OL}	—	—	—	0.2VDD	V
Crystal out (high)	V_{OH}	—	0.8VDD	—	—	V
External load cap	R_{FB}		6	7	12.5	pF
Feedback resistance	C_L		5.5	11.2	22.8	MΩ

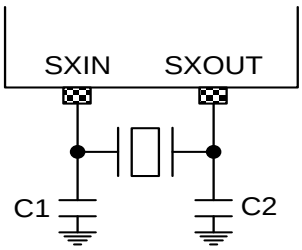


Figure 179. Crystal Oscillator

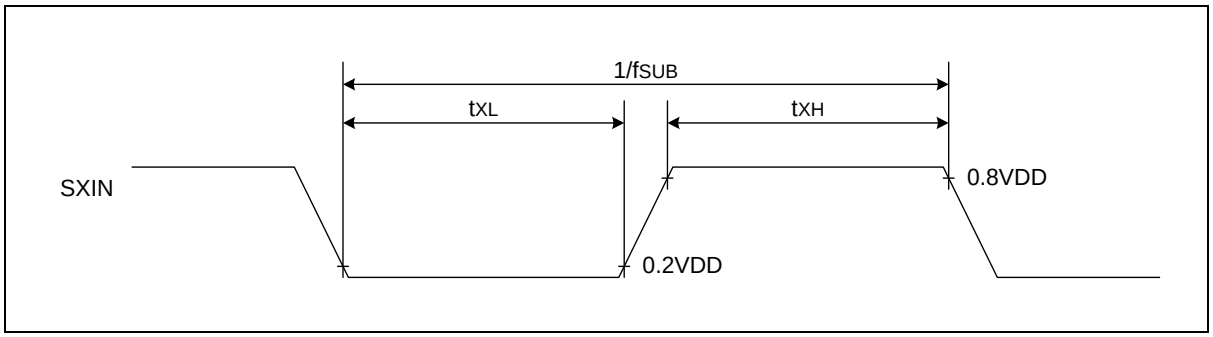


Figure 180. Clock Timing Measurement at SXIN

25.19 Operating voltage range

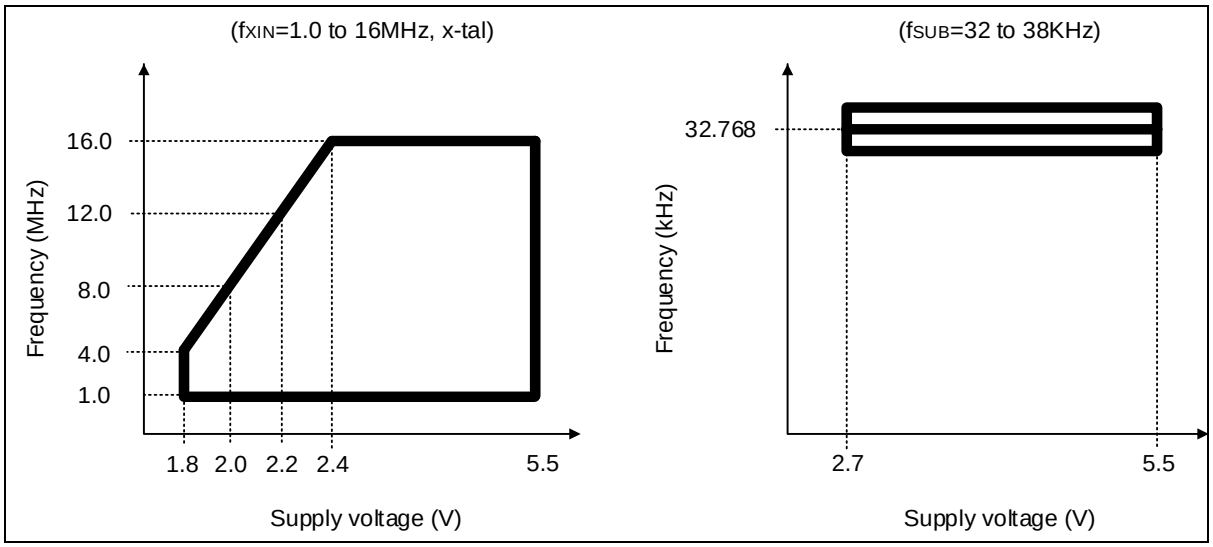


Figure 181. Operating Voltage Range

25.20 PLL electrical characteristics

Table 119. PLL Electrical Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		1.8	5	5.5	V
Operating current	I_{DD}	Enable	—	—	1	mA
		Disable	—	5	500	nA
Output frequency	f_{OUT}	—	—	—	144	MHz
Duty	f_{DUTY}		40	—	60	%
Input frequency	$f_{PLLINCLK}$		4	8	16	MHz
Locking time*	t_{LOCK}		190	—	—	us

25.21 Comparator characteristics

The measured value for the parameters and conditions listed below were confirmed by simulaion

Table 120. Comparator Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		2.0	5	5.5	V
Comparator offset error	V_{Offset}		—	±5	±10	mV
		Using internal VREF		±20	±70	
Hysteresis	$Vhys_{(None)}$	—	—	0		mV
	$Vhys_{(Low)}$	High speed mode	3	8	13	
		All other power modes	5	8	10	
	$Vhys_{(Mid)}$	High speed mode	7	15	26	
		All other power modes	9	15	19	
	$Vhys_{(High)}$	High speed mode	18	31	49	
		All other power modes	19	31	40	
Propagation delay for 200mv Step with 100mv overdrive	t_{PD}	Ultra-low power mode		2	4.5	us
		Low power mode		0.7	1.5	
		Medium power mode		0.3	0.6	
		High speed mode VDDEXT≥2.7V		50	100	ns
		High speed mode 2V<VDDEXT<2.7V		100	240	
Propagation delay for full Range step with 100mv Overdrive	t_{PD}	Ultra-low power mode		2	7	us
		Low power mode		0.7	2.1	
		Medium power mode		0.3	1.2	
		High speed mode VDDEXT≥2.7V		90	180	ns
		High speed mode 2V<VDDEXT<2.7V		110	300	
Comparator current	$IDD_{(COMP)}$	Ultra-low power mode (COMP1 only)	—	1.2	1.5	uA

Table 120. Comparator Characteristics (continued)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Comparator current	$IDD_{(COMP)}$	Ultra-low power mode (COMP1 and COMP2)	—	2.4	—	uA
		Low power mode (COMP1 only)	—	3	5	
		Low power mode (COMP1 and COMP2)	—	6	—	
		Medium power mode (COMP1 only)	—	10	15	
		Medium power mode (COMP1 and COMP2)	—	20	—	
		High Speed mode (COMP1 only)	—	75	100	
		High Speed mode (COMP1 and COMP2)	—	150	—	

Table 121. Interanal Voltage Reference Characteristics(For comparator)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		2.7	5	5.5	V
Output Voltage	$V_{OUT1.2V}$		1.162	1.223	1.284	
	$V_{OUT0.9V}$		0.871	0.917	0.963	
	$V_{OUT0.6V}$		0.580	0.611	0.642	
	$V_{OUT0.3V}$		0.290	0.305	0.320	

25.22 USB characteristics

Table 122. USB Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		3.0	-	5.5	V
	VDD33		2.97	3.3	3.63	V
	VDDIO2		2.97	3.3	3.63	
Pull down resistance on DP/DN	R_{PD}	Enable internal resistor	12	-	17	KΩ
Pull up resistance on DP/DN	R_{PU}	Enable internal resistor	1.1	-	1.9	KΩ
Output driver impedance	Z_{DRV}	Driving high and low	10	-	35	Ω

25.23 Temperature Sensor characteristics

Table 123. Temperature Sensor Characteristics

(Temperature: -40℃ to +85℃)

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating Voltage	VDD		1.8	-	5.5	V
Temperature Accuracy error	$T_{SENSACC}$			±10		℃

The measured value for the parameters and conditions listed below were confirmed by simulaion

26. Development tools

This chapter introduces wide range of development tools for A31G32x. ABOV offers software tools, debuggers, and programmers to help a user in generating right results to match target applications. ABOV supports entire development ecosystem of the customers.

26.1 Compiler

ABOV semiconductor does not provide any compiler for A31G32x. However, since A31G32x have ARM's high-speed 32-bit Cortex-M0+ Cores for their CPU, you can use all kinds of third party's standard compiler such as Keil C Compiler. These compilers' output debug information can be integrated with our A-Link and A-Link Pro. Please visit our website www.abovsemi.com for more information regarding the A-Link and A-Link Pro.

26.2 Debugger

The A-Link and A-Link Pro support ABOV Semiconductor's A31G32x MCU emulation in SWD Interface. The A-Link and A-Link Pro use two wires interfacing between PC and MCU, which is attached to user's system. The A-Link and A-Link Pro can read or change the value of MCU's internal memory and I/O peripherals. In addition, the A-Link and A-Link Pro control MCU's internal debugging logic. This means A-Link and A-Link Pro control emulation, step run, monitoring and many more functions regarding debugging.

The A-Link and A-Link Pro run underneath MS operating system such as MS-Windows NT/ 2000/ XP/ Vista/ 7/ 8/ 8.1/ 10 (32-bit, 64-bit).

Programming information using the A-Link and A-Link Pro are provided in Figure 185. More detailed information about the A-Link and A-Link Pro, please visit our website www.abovsemi.com and download the debugger S/W and documents.

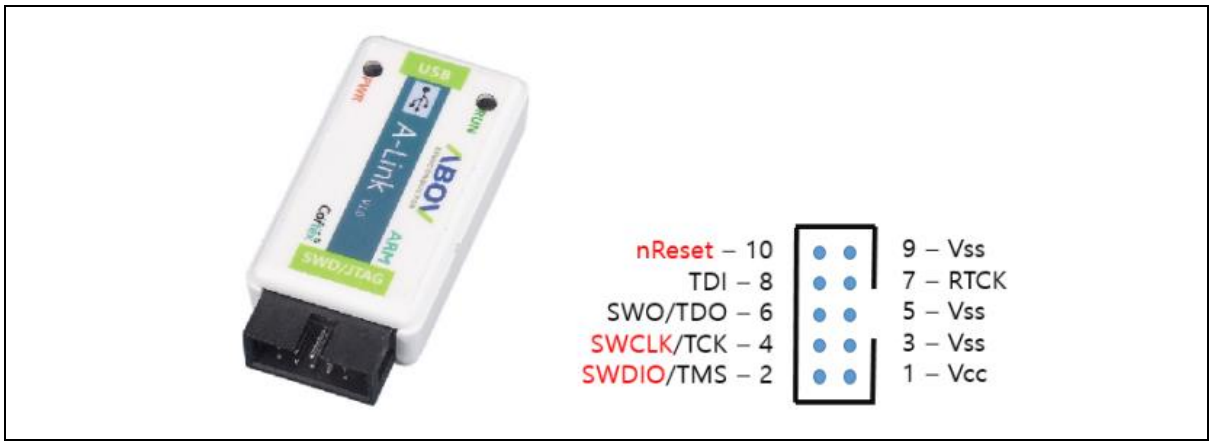


Figure 182. A-Link and Pin Descriptions

26.3 Programmer

E-PGM+

E-PGM+ is a single programmer, and allows a user to program on the device directly.

- Support ABOV devices
- 2~5 times faster than S-PGM+
- Main controller: 32-bit MCU @ 72MHz
- Buffer memory: 1MB

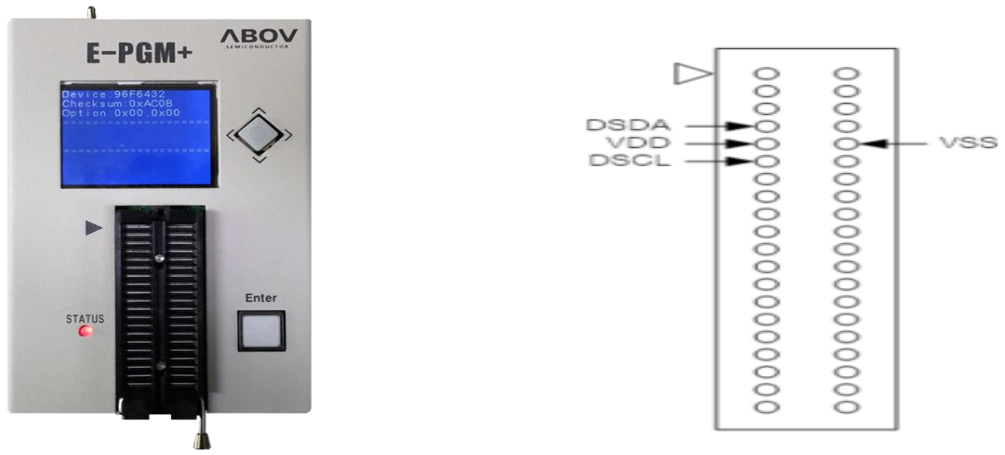


Figure 183. E-PGM+ (Single Writer) and Pin Descriptions

Gang programmer

E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.

Figure 184. E-Gang4 and E-Gang6 (for Mass Production)

27. Circuit Design Guide

Refer to the Recommended Circuit and Layout Design as shown below.

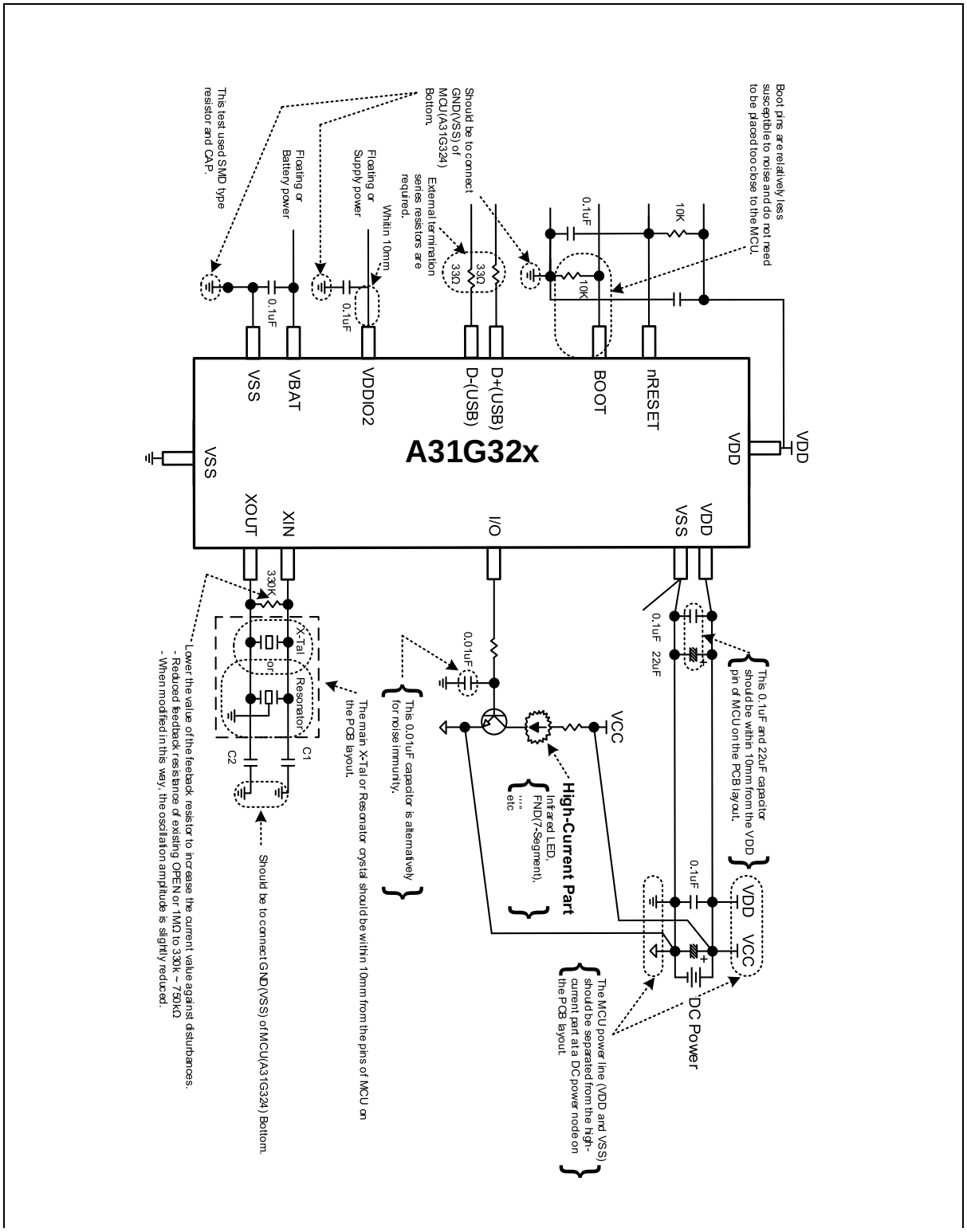


Figure 185. PCB Design Guide for On-Board Programming

28. Package information

This chapter provides A31G32x series package information.

28.1 64 LQFP package information

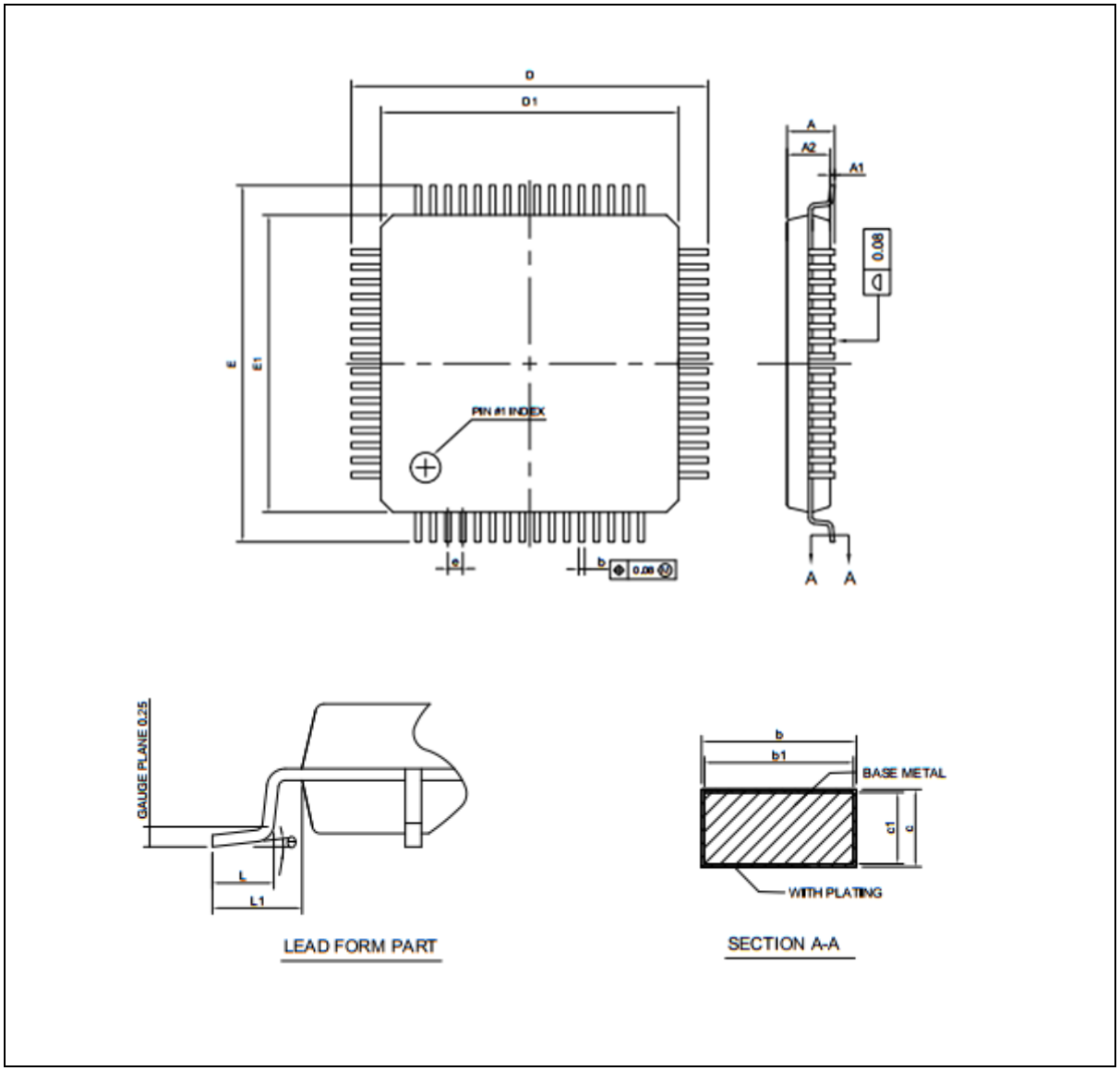


Figure 186. 64 LQFP Package Outline

Table 124. 64 LQFP Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
b1	0.17	0.20	0.23
c	0.09	—	0.20
c1	0.09	—	0.16
D	11.80	12.00	12.20
D1	9.80	10.00	10.20
E	11.80	12.00	12.20
E1	9.80	10.00	10.20
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
Θ	0°	3.5°	7°

NOTES:

1. All dimension refer to JEDEC standard MS-026-BCD.
2. Dimensions 'D1' and 'E1' do not include MOLD PROTRUSION. Allowable PROTRUSION is 0.25 mm per side. 'D1' and 'E1' are maximum plastic body size dimensions including MOLD MISMATCH.
3. Dimension 'b' does not include DAMBAR PROTRUSION. Allowable DAMBAR PROTRUSION shall not cause the LEAD width to exceed the maximum 'b' dimension by more than 0.08 mm.
4. 'A1' is defined as the distance from the seating plane to the lowest point on the package body.

28.2 48 LQFP package information

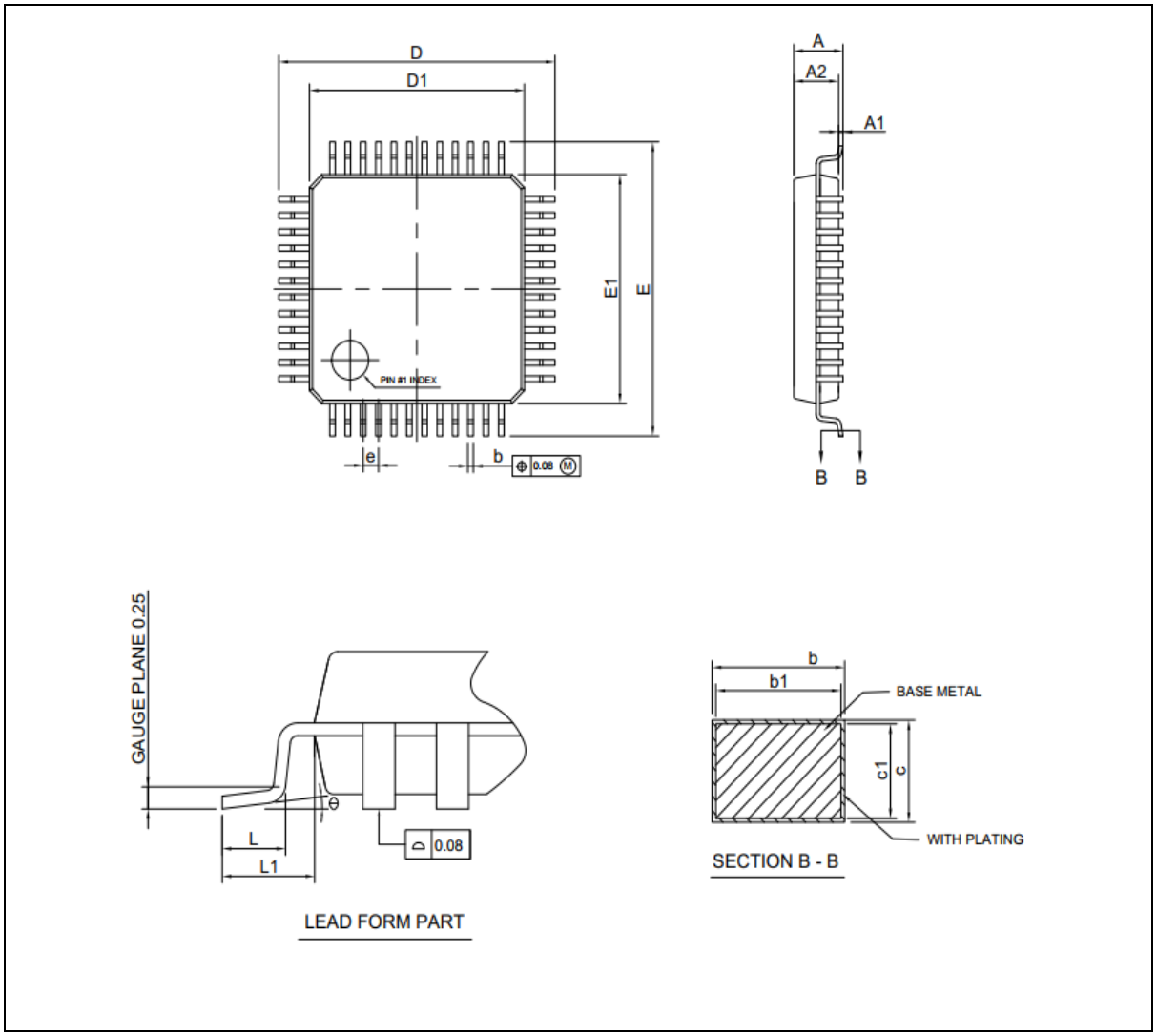


Figure 187. 48 LQFP Package Outline

Table 125. 48 LQFP Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
b1	0.17	0.20	0.23
c	0.09	—	0.20
c1	0.09	—	0.16
D	8.80	9.00	9.20
D1	6.80	7.00	7.20
E	8.80	9.00	9.20
E1	6.80	7.00	7.20
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
Θ	0°	3.5°	7°

NOTES:

1. All dimension refer to JEDEC standard MS-026-BBC.
2. Dimensions 'D1' and 'E1' do not include MOLD PROTRUSION. Allowable PROTRUSION is 0.25 mm per side. 'D1' and 'E1' are maximum plastic body size dimensions including MOLD MISMATCH.
3. Dimension 'b' does not include DAMBAR PROTRUSION. Allowable DAMBAR PROTRUSION shall not cause the LEAD width to exceed the maximum 'b' dimension by more than 0.08 mm.
4. 'A1' is defined as the distance from the seating plane to the lowest point on the package body.

28.3 48 QFN package information

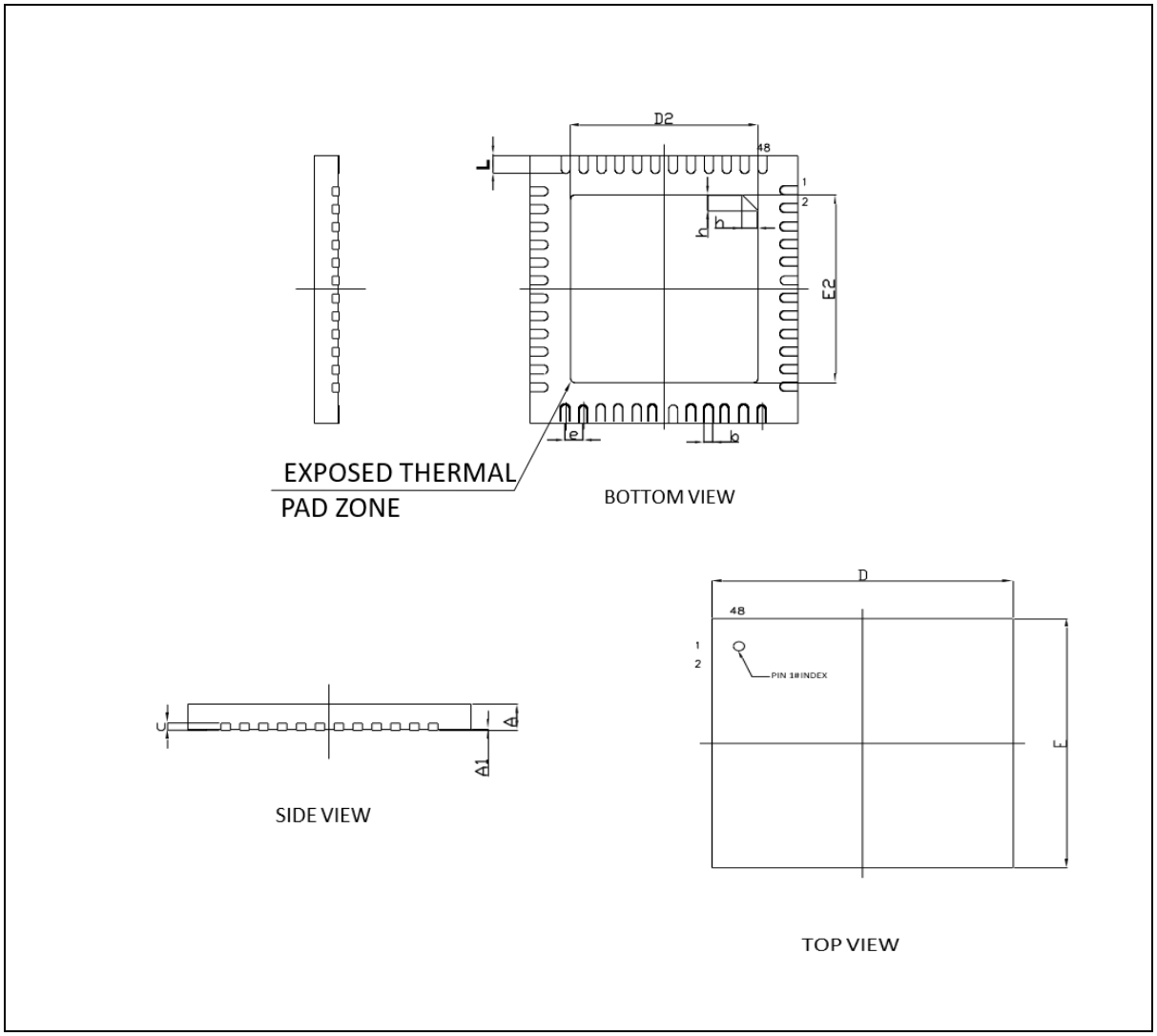


Figure 188. 48 QFN Package Outline

Table 126. 48 QFN Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	0.50	0.55	0.60
A1	0.00	0.02	0.05
b	0.10	0.15	0.20
c	0.10	0.15	0.20
D	4.90	5.00	5.10
D2	3.60	3.70	3.80
E	4.90	5.00	5.10
E2	3.60	3.70	3.80
e	0.35 BSC		
L	0.30	0.35	0.40
h	0.30	0.35	0.40

NOTES:

1. Dimension 'b' applies to metallized terminal and is measured between 0.15mm and 0.3mm from the TERMINAL TIP. If the TERMINAL has the optinal radius on the other end of the TERMINAL, the Dimension 'b' should not be measured in that radius area.

29. Ordering information

Table 127. A31G32x Device Ordering Information

Device name	Flash	SRAM	SPI	USART	I2C	Timer	PWM	ADC	I/O ports	Package
A31G324RLN	128KB	16KB	2	4	2	6(16-bit)/2(32-bit)	8	16	51	LQFP 64
A31G324CLN*	128KB	16KB	2	3	2	6(16-bit)/2(32-bit)	8	10	37	LQFP 48
A31G324CUN*	128KB	16KB	2	3	2	6(16-bit)/2(32-bit)	8	10	37	QFN 48
A31G323RLN*	64KB	16KB	2	4	2	6(16-bit)/2(32-bit)	8	16	51	LQFP 64
A31G323CLN*	64KB	16KB	2	3	2	6(16-bit)/2(32-bit)	8	10	37	LQFP 48
A31G323CUN*	64KB	16KB	2	3	2	6(16-bit)/2(32-bit)	8	10	37	QFN 48

* For available options or further information on the devices with "*" marks, please contact the ABOV Sales Office

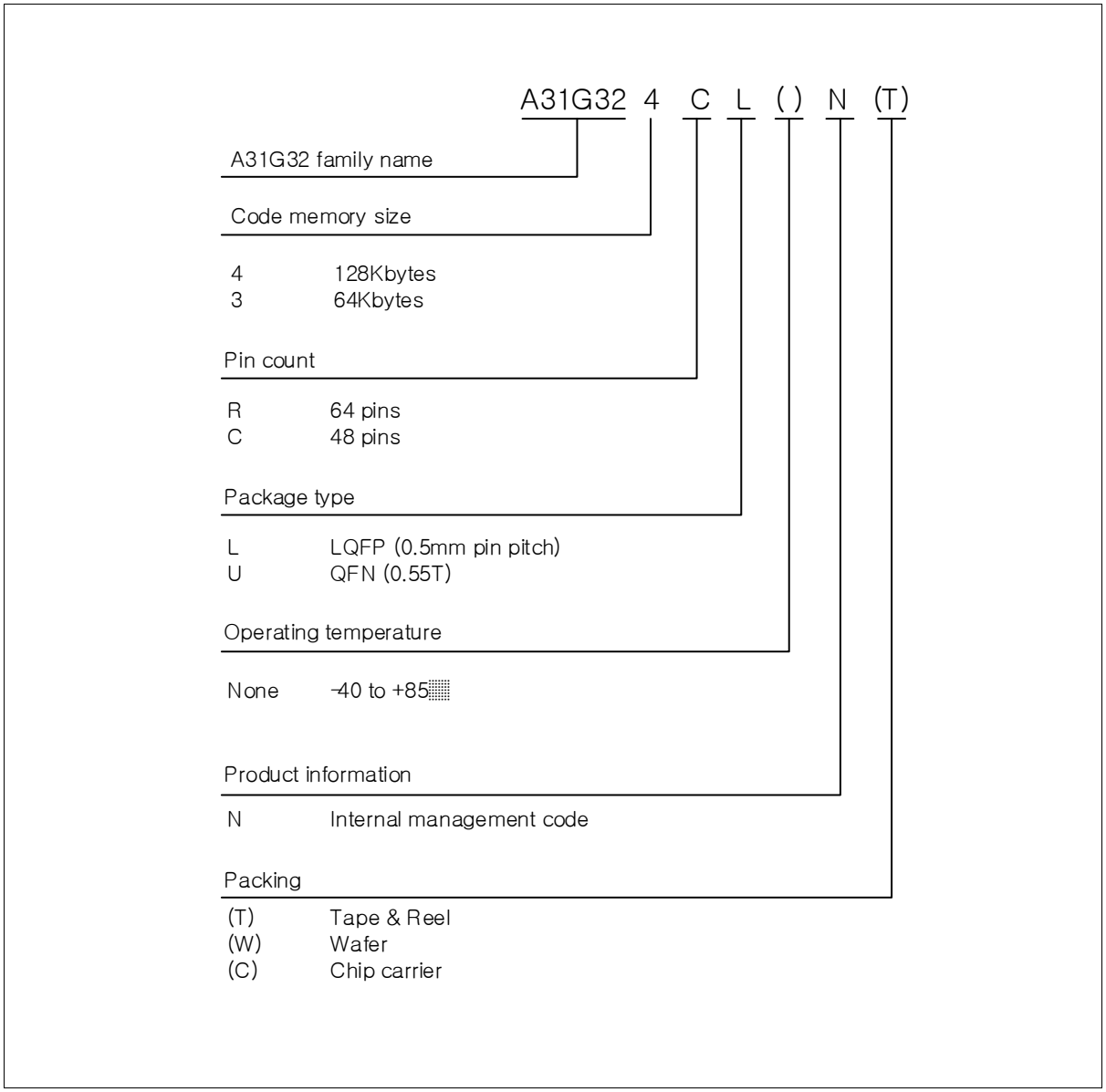


Figure 189. A31G32x Device Numbering Nomenclature

Revision history

Date	Revision	Description
Dec. 04, 2020	1.00	First creation
Jan. 08, 2021	1.01	Add caution statement to ADC CCR register
Jan. 11, 2021	1.02	Updated Introduction
Feb. 23, 2021	1.03	Updated Recommended operating conditions and Main oscillator characteristics Updated How to calculate and convert to temperature
Nov. 1, 2021	1.04	Updated caution statement to SPI BR register
Apr. 18, 2022	1.05	Removed Industrial Grade Updated Ordering information Added the Figure 188. 48 QFN Package Outline Added the Figure 4. QFN 48 Pinouts
Aug. 19, 2022	1.06	Updated the Figure 4. QFN 48 Pinouts Updated the 14.2.6
Oct. 14, 2022	1.07	Revised the font of this document
Dec. 2, 2024	1.08	Updated the disclaimer.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
93, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr

ABOV Disclaimer

IMPORTANT NOTICE – PLEASE READ CAREFULLY

ABOV Semiconductor ("ABOV") reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. **ABOV DOES NOT GIVE WARRANTIES AS TO THE ACCURACY OR COMPLETENESS OF THE INFORMATION INCLUDED HEREIN.** Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. **NO LICENSE, EXPRESS OR IMPLIED, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY ABOV HEREIN. ABOV DISCLAIMES ALL EXPRESS AND IMPLIED WARRANTIES AND SHALL NOT BE RESPONSIBLE OR LIABLE FOR ANY INJUIRES OR DAMAGES RELATED TO USE OF ABOV PRODUCTS IN SUCH UNAUTHORIZED APPLICATIONS.** ABOV and the ABOV logo are trademarks of ABOV. For additional information about ABOV trademarks, please refer to https://www.abov.co.kr/en/about/corporate_identity.php. All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.

© 2020 ABOV Semiconductor – All rights reserved