A33M116, A33M114
User's Manual

32-bit Cortex-M3 based High-performance Microcontroller
Flash 256KB, Data Flash 32KB, SRAM 16KB
Advanced Motor Control PWM

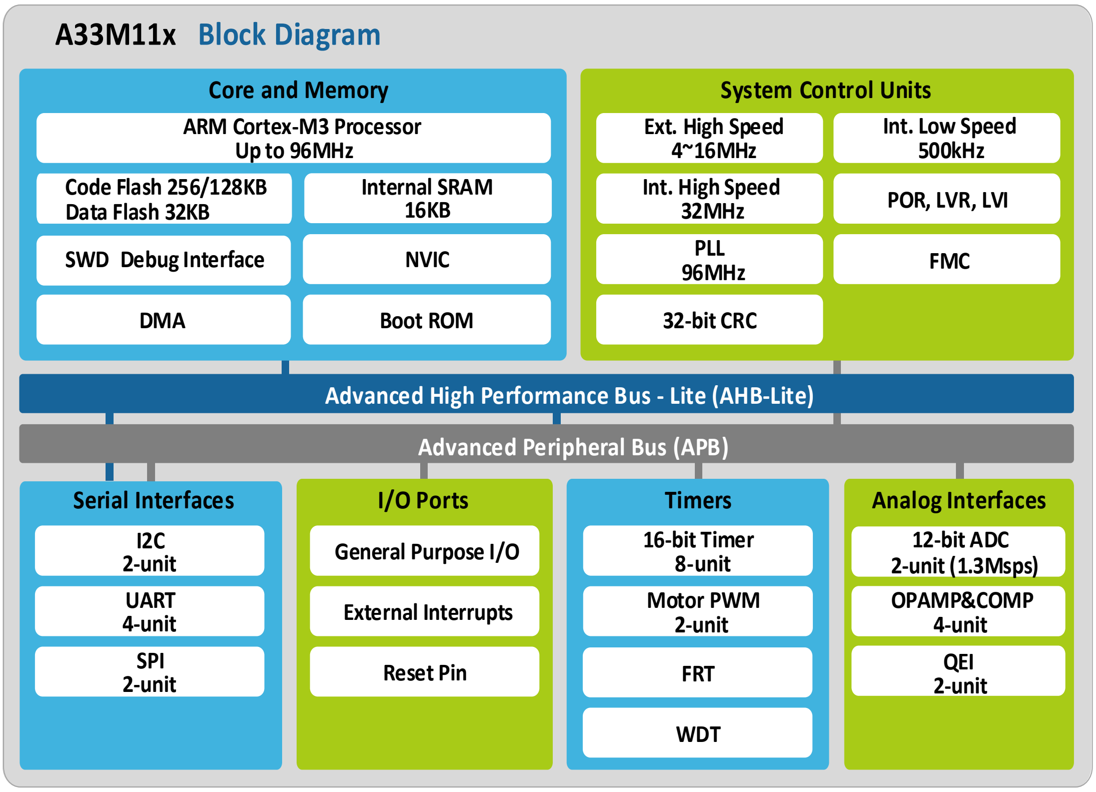
UM Rev. 1.05

Introduction

A33M11x series is a 32-bit microcontroller unit (MCU) that is suitable for applications requiring high-performance processing such as electric motors. It includes a high performance 32-bit core, ARM Cortex-M3, and various peripheral devices for motor control.

Two 3-phase PWM generators are able to control two inverter motors simultaneously, and go into an individual mode when it is optimized for Induction Heating (IH). Two high-speed 12-bit ADCs take care of motor driving information coming in through 22 analog channels. In addition, A33M11x can control either two inverter motors, or one inverter motor and a power factor correction (PFC) process, simultaneously. It can communicate with external devices via various interfaces such as UART, SPI and I2C modules.

Figure 1. A33M11x Concept Block Diagram



Reference document

- Document 'DDI337' is provided by ARM and contains information of Cortex-M3.

Contents

Introduction 1
Reference document 1
1 Description 19
1.1 Device overview 19
1.2 A33M11x block diagram 24
2 Pinouts and pin descriptions 25
2.1 Pinouts 25
2.1.1 A33M116RL/A33M116RM/A33M114RL (LQFP-64) 25
2.1.2 A33M116CL/A33M114CL (LQFP-48) 26
2.1.3 A33M114SN (LQFP-44) 27
2.2 Pin description 28
3 System and memory overview 33
3.1 System architecture 33
3.1.1 Cortex-M3 core 33
3.1.2 Interrupt controller 33
3.2 Memory organization 37
3.2.1 Register boundary address 37
3.2.2 Memory map 38
3.2.3 Embedded SRAM 39
3.2.4 Flash memory overview 39
3.2.5 Boot mode 39
3.2.5.1 Boot mode pins 39
3.2.5.2 Boot mode connections 40
3.2.5.3 SWD mode connections 40
4 System Control Unit (SCU) 41
4.1 SCU block diagram 41
4.2 Clock system 42
4.2.1 Configuration of miscellaneous clocks 43
4.2.2 HCLK clock domain 45
4.2.3 PCLK clock domain 45
4.2.4 Clock configuration procedure 45
4.3 Reset 50
4.3.1 Cold reset 51
4.3.2 Warm reset 52
4.3.3 LVR reset 52
4.3.4 Reset tree 53
4.4 Operation mode 54
4.4.1 RUN mode 55
4.4.2 SLEEP mode 55
4.4.3 STOP mode 56
4.5 Registers 59
4.5.1 CHIPCONFIG_VENDORID: vendor ID register 61
4.5.2 CHIPCONFIG_CHIPID: chip ID register 61

4.5.3 CHIPCONFIG_REVNR: revision number register 61
4.5.4 SCU_SMR: system mode register 62
4.5.5 SCU_SRCR: system reset control register 63
4.5.6 SCU_WUER: wakeup source enable register 64
4.5.7 SCU_WUSR: wakeup source status register 65
4.5.8 SCU_RSER: reset source enable register 66
4.5.9 SCU_RSSR: reset source status register 67
4.5.10 SCU_PRER1: peripheral reset enable register 1 69
4.5.11 SCU_PRER2: peripheral reset enable register 2 70
4.5.12 SCU_PER1: peripheral enable register1 71
4.5.13 SCU_PER2: peripheral enable register 2 72
4.5.14 SCU_PCER1: peripheral clock enable register 1 73
4.5.15 SCU_PCER2: peripheral clock register 2 74
4.5.16 SCU_CSCR: clock source control register 75
4.5.17 SCU_SCCR: system clock control register 76
4.5.18 SCU_CMR: clock monitoring register 78
4.5.19 SCU_COR: clock output register 79
4.5.20 SCU_NMICR: NMI control register 80
4.5.21 SCU_NMISR: NMI status register 81
4.5.22 SCU_PLLCON: PLL control register 82
4.5.23 SCU_VDCCON: VDC control register 84
4.5.24 SCU_LVICR: LVI (Low-Voltage Indicator) control register 85
4.5.25 SCU_LVISR: LVI (Low-Voltage Indicator) status register 86
4.5.26 SCU_LVRCR: LVR (Low-Voltage Reset) control register 87
4.5.27 SCU_EOSCR: external oscillator control register 88
4.5.28 SCU_MCCR1: miscellaneous clock control register 1 89
4.5.29 SCU_MCCR2: miscellaneous clock control register 2 90
4.5.30 SCU_MCCR3: miscellaneous clock control register 3 91
4.5.31 SCU_MCCR4: miscellaneous clock control register 4 92
4.5.32 SCU_MCCR5: miscellaneous clock control register 5 93
4.5.33 SCU_MCCR6: miscellaneous clock control register 6 94
4.5.34 SCU_MCCR7: miscellaneous clock control register 7 95
4.5.35 SCU_SYSTEN: system access enable register 96
4.6 Functional description 97
4.6.1 PLL clock setting 97
4.6.2 Clock monitoring 98
4.6.3 Setting examples 100
5 Port Control Unit (PCU) and GPIO 101
5.1 PCU and GPIO block diagram 103
5.2 Pin multiplexing 106
5.3 Registers 108
5.3.1 Pn_MR1: PORT n mux1 select register 110
5.3.2 Pn_MR2: PORT n mux2 select register 111
5.3.3 Pn_CR: PORT n control register 112
5.3.4 Pn_PRCR: PORT n pull-up resistor control register 113

5.3.5 Pn_DER: PORT n debouncing enable register ... 113
5.3.6 Pn_STR: PORT n strength configuration register ... 114
5.3.7 Pn_IER: PORT n interrupt enable register ... 115
5.3.8 Pn_ISR: PORT n interrupt status register ... 116
5.3.9 Pn_ICR: PORT n interrupt control register ... 117
5.3.10 Pn_ODR: PORT n output data register ... 117
5.3.11 Pn_IDR: PORT n input data register ... 118
5.3.12 Pn_BSR: PORT n set/reset register ... 119
5.3.13 Pn_BCR: PORT n reset register ... 119
5.3.14 PORTEN: PORT access enable register ... 120
5.4 Functional description ... 121
5.4.1 Port functionality ... 121
5.4.2 Debouncing functionality ... 122
5.4.3 I/O block functionality ... 123
5.4.4 Interrupt functionality ... 124
5.4.5 Setting examples ... 125
6 Flash Memory Controller (FMC) ... 126
6.1 Code Flash registers ... 129
6.1.1 CFMC_CONF: code Flash control register ... 130
6.1.2 CFMC_FLSKEY: code Flash access key register ... 132
6.1.3 CFMC_OTPKEY: code Flash OTP access key register ... 132
6.1.4 CFMC_FLSPROT: code Flash protection register ... 133
6.1.5 CFMC_OTPPROT: code Flash OTP protection register ... 134
6.1.6 CFMC_CTRL: code Flash access control register ... 135
6.1.7 CFMC_STAT: code Flash access status register ... 136
6.1.8 CFMC_READPROT: code Flash read protection register ... 137
6.1.9 CFMC_PWIN: code Flash password input register ... 139
6.1.10 CFMC_CHKCTRL: code Flash checksum control register ... 140
6.1.11 CFMC_CHKDOUT: code Flash checksum data output register ... 140
6.1.12 CFMC_CHKSADDR: code Flash checksum start address register ... 141
6.1.13 CFMC_CHKEADDR: code Flash checksum end address register ... 141
6.1.14 CFMC_WTIMEOUT: code Flash write timeout register ... 142
6.1.15 CFMC_PWPRST: code Flash password preset register ... 142
6.2 Data Flash registers ... 143
6.2.1 DFMC_CONF: data Flash control register ... 144
6.2.2 DFMC_FLSKEY: data Flash access key register ... 145
6.2.3 DFMC_FLSPROT: data Flash protection register ... 146
6.2.4 DFMC_CTRL: data Flash access control register ... 147
6.2.5 DFMC_STAT: data Flash access status register ... 148
6.2.6 DFMC_CHKCTRL: data Flash checksum control register ... 149
6.2.7 DFMC_CHKDOUT: data Flash checksum data output register ... 149
6.2.8 DFMC_CHKSADDR: data Flash checksum start address register ... 150
6.2.9 DFMC_CHKEADDR: data Flash checksum end address register ... 150
6.2.10 DFMC_WTIMEOUT: data Flash write timeout register ... 151
6.3 Functional description ... 152

6.3.1 How to lock/unlock 152
6.3.2 Sequence of resetting all caches 152
6.3.3 Writing sequence 152
6.3.3.1 Flash program example 153
6.3.3.2 Flash burst program example 154
6.3.3.3 Flash page erase example 155
6.3.3.4 Flash 2KB sector erase example 156
6.3.3.5 Chip erase example 157
6.3.4 Checksum control sequence 158
6.3.5 Setting examples 159
7 Direct Memory Access Controller (DMAC) 160
7.1 DMAC block diagram 160
7.2 Registers 161
7.2.1 DMAn_CR: DMA controller register 162
7.2.2 DMAn_SR: DMA status register 163
7.2.3 DMAn_PAR: DMA n peripheral device address register 163
7.2.4 DMAn_MAR: DMA n memory address register 164
7.3 Functional description 165
7.3.1 DMA transfer timing from a peripheral to a memory 166
7.3.2 DMA transfer timing from a memory to a peripheral 167
7.3.3 DMA transfer 167
7.3.4 Setting examples 168
8 Watchdog Timer (WDT) 169
8.1 WDT block diagram 170
8.2 Registers 171
8.2.1 WDT_LR: watchdog timer load register 171
8.2.2 WDT_CNT: watchdog current count register 172
8.2.3 WDT_CON: watchdog control register 173
8.2.4 WDT_AEN: watchdog access enable register 174
8.3 Functional description 175
8.3.1 WDT down-counter control 175
8.3.2 WDT reset mode 175
8.3.3 WDT interrupt mode 175
8.3.4 Using the WDT in debug mode 175
8.3.5 Timing diagram 176
8.3.6 Prescaler table 176
8.3.7 Setting examples 177
9 16-bit timer 178
9.1 16-bit timer block diagram 180
9.2 Registers 181
9.2.1 TIMERn_CR1: timer/counter n control register 1 182
9.2.2 TIMERn_CR2: timer/counter n control register 2 184
9.2.3 TIMERn_PRS: timer/counter n prescaler register 184
9.2.4 TIMERn_GRA: timer/counter n general data register A 185
9.2.5 TIMERn_GRB: timer/counter n general data register B 186

9.2.6 TIMERn_CNT: timer/counter n count register 187
9.2.7 TIMERn_SR: timer/counter n status register 187
9.2.8 TIMERn_IER: timer/counter n interrupt enable register 188
9.2.9 TIMERn_TRGPNT: timer/counter n trigger point register 188
9.2.10 TIMERn_SYNC: timer/counter n sync configuration register 189
9.3 Functional description 190
9.3.1 Basic timer operations 190
9.3.2 Normal periodic mode 191
9.3.3 One-shot mode 193
9.3.4 PWM timer output 194
9.3.5 PWM synchronization 195
9.3.6 PWM delayed synchronization 197
9.3.7 Capture mode 198
9.3.8 ADC triggering 199
9.3.9 Setting examples 201
10 Free Run Timer (FRT) 202
10.1 FRT block diagram 202
10.2 Registers 203
10.2.1 FRT_CTRL: FRT control register 203
10.2.2 FRT_MCNT: FRT match counter register 204
10.2.3 FRT_CNT: FRT counter register 204
10.2.4 FRT_STAT: FRT status register 205
10.3 Functional description 206
10.3.1 Match interrupt operation 206
10.3.2 Overflow interrupt operation 207
10.3.3 Setting examples 207
11 Universal Asynchronous Receiver/Transmitter (UART) 208
11.1 UART block diagram 210
11.2 Registers 211
11.2.1 UARTn_RBR: UART n receive data buffer register 212
11.2.2 UARTn_THR: UART n transmit data hold register 212
11.2.3 UARTn_IER: UART n interrupt enable register 213
11.2.4 UARTn_IIR: UART n interrupt ID register 214
11.2.5 UARTn_LCR: UART n line control register 216
11.2.6 UARTn_DCR: UART n data control register 218
11.2.7 UARTn_LSR: UART n line status register 219
11.2.8 UARTn_BDR: UART n baud-rate divisor latch register 221
11.2.9 UARTn_BFR: UART n baud-rate fraction counter register 222
11.2.10 UARTn_IDTR: UART n inter-frame delay time register 223
11.3 Functional description 224
11.3.1 Receive data sampling timing 224
11.3.2 Transmit data format 225
11.3.3 Transmit interrupt 225
11.3.4 Multi-sampling 226
11.3.5 Start bit detection 227

11.3.6 Data sampling strategy 228
11.3.7 Inter-frame delay for data transmission 229
11.3.8 DMA transfers 229
11.3.9 Setting examples 230
12 Serial Peripheral Interface (SPI) 231
12.1 SPI block diagram 233
12.2 Registers 234
12.2.1 SPIn_TDR: SPI n transmit data register 234
12.2.2 SPIn_RDR: SPI n receive data register 235
12.2.3 SPIn_CR: SPI n control register 236
12.2.4 SPIn_SR: SPI n status register 238
12.2.5 SPIn_BR: SPI n baud rate register 239
12.2.6 SPIn_EN: SPI n enable register 239
12.2.7 SPIn_LR: SPI n delay length register 240
12.3 Functional description 241
12.3.1 Slave selection pin 241
12.3.2 Clock phase and clock polarity 241
12.3.3 DMA handshake 244
12.3.4 Setting examples 245
13 Inter Integrated Circuit (I2C) 248
13.1 I2C block diagram 250
13.2 Registers 251
13.2.1 I2Cn_DR: I2Cn data register 251
13.2.2 I2Cn_SR: I2Cn status register 252
13.2.3 I2Cn_SAR: I2Cn slave address register 253
13.2.4 I2Cn_CR: I2Cn control register 254
13.2.5 I2Cn_SCLLR: I2Cn SCL low duration register 255
13.2.6 I2Cn_SCLHR: I2Cn SCL high duration register 256
13.2.7 I2Cn_SDAHR: I2Cn SDA hold register 257
13.3 Functional description 258
13.3.1 I2C bit transfer 258
13.3.2 START/repeated START/STOP 258
13.3.3 Data transfer 259
13.3.4 Acknowledge 260
13.3.5 Synchronization 261
13.3.6 Arbitration 262
13.3.7 I2C operation 263
13.3.7.1 Master transmitter 263
13.3.7.2 Master receiver 266
13.3.7.3 Slave transmitter 269
13.3.7.4 Slave receiver 271
13.3.8 Setting examples 273
14 Motor Pulse Width Modulation (MPWM) 274
14.1 MPWM block diagram 276
14.2 Registers 279

14.2.1 MPWMn_MR: MPWM n mode register 281
14.2.2 MPWMn_OLR: MPWM n output level register 283
14.2.3 MPWMn_FOLR: MPWM n forced output level register 285
14.2.4 MPWMn_PRD: MPWM n period register 286
14.2.5 MPWMn_DUH: MPWM n duty UH register 286
14.2.6 MPWMn_DVH: MPWM n duty VH register 287
14.2.7 MPWMn_DWH: MPWM n duty WH register 287
14.2.8 MPWMn_DUL: MPWM n duty UL register 288
14.2.9 MPWMn_DVL: MPWM n duty VL register 288
14.2.10 MPWMn_DWL: MPWM n duty WL register 289
14.2.11 MPWMn_CR1: MPWM n control register 1 289
14.2.12 MPWMn_CR2: MPWM n control register 2 290
14.2.13 MPWMn_SR: MPWM n status register 291
14.2.14 MPWMn_IER: MPWM n interrupt enable register 293
14.2.15 MPWMn_CNT: MPWM n counter register 295
14.2.16 MPWMn_DTR: MPWM n dead time register 2 296
14.2.17 MPWMn_PCR: MPWM n protection control register 297
14.2.18 MPWMn_PSR: MPWM n protection status register 299
14.2.19 MPWMn_OCR: MPWM n overvoltage control register 300
14.2.20 MPWMn_OSR: MPWM n overvoltage status register 302
14.2.21 MPWMn_ATRm: MPWM n ADC trigger counter register 303
14.2.22 MPWMn_CPCR: MPWM n comparator protection control register 304
14.3 Individual mode registers 306
14.3.1 MPWMn_CR3: MPWM n control register 3 306
14.3.2 MPWMn_CR4: MPWM n control register 4 308
14.3.3 MPWMn_PRDU: MPWM n phase U period register 310
14.3.4 MPWMn_PRDV: MPWM n phase V period register 310
14.3.5 MPWMn_PRDW: MPWM n phase W period register 311
14.3.6 MPWMn_CNTU: MPWM n phase U period register 311
14.3.7 MPWMn_CNTV: MPWM n phase V period register 312
14.3.8 MPWMn_CNTW: MPWM n phase W period register 312
14.3.9 MPWMn_DTRU: MPWM n phase U dead time register 313
14.3.10 MPWMn_DTRV: MPWM n phase V dead time register 315
14.3.11 MPWMn_DTRW: MPWM n phase W dead time register 317
14.3.12 MPWMn_CAPCNTx: MPWM n phase U/V/W capture counter register 319
14.3.13 MPWMn_RCAPx: MPWM n phase U/V/W capture rising value register 320
14.3.14 MPWMn_FCAPx: MPWM n phase U/V/W capture falling value register 321
14.3.15 MPWMn_SCAPx: MPWM n phase U/V/W sub-capture value register 322
14.4 Functional description 323
14.4.1 Normal PWM up-count mode timing 324
14.4.2 Normal PWM up-down count mode timing 326
14.4.3 MPWM two-channel symmetric mode timing 327
14.4.4 Motor PWM one-channel asymmetric mode timing 328
14.4.5 Motor PWM one-channel symmetric mode timing 329
14.4.6 Individual PWM two-channel symmetric mode timing 330

14.4.7 Individual PWM one-channel asymmetric mode timing ... 331
14.4.8 Individual PWM one-channel symmetric mode timing ... 332
14.4.9 MPWM dead time operation ... 333
14.4.10 Special case examples of MPWM dead time timing ... 336
14.4.11 Symmetric mode vs. asymmetric mode ... 343
14.4.12 Functional description of protection and overvoltage ... 345
14.4.13 Functional description of ADC triggers ... 346
14.4.14 Interrupt generation timing ... 348
14.4.15 Setting examples ... 349
15 Quadrature Encoder Interface (QEI) ... 351
15.1 QEI block diagram ... 353
15.2 Registers ... 354
15.2.1 QEIn_MR: QEI n mode register ... 355
15.2.2 QEIn_CON: QEI n control register ... 356
15.2.3 QEIn_SR: QEI n status register ... 356
15.2.4 QEIn_POS: QEI n position counter register ... 357
15.2.5 QEIn_MAX: QEI n maximum position register ... 357
15.2.6 QEIn_CMP0: QEI n position compare 0 register ... 358
15.2.7 QEIn_CMP1: QEI n position compare 1 register ... 358
15.2.8 QEIn_CMP2: QEI n position compare 2 register ... 359
15.2.9 QEIn_IDX: QEI n index counter register ... 359
15.2.10 QEIn_CMPI: QEI n index compare register ... 360
15.2.11 QEIn_VLR: QEI n velocity reload register ... 360
15.2.12 QEIn_VLT: QEI n velocity timer register ... 361
15.2.13 QEIn_VLP: QEI n velocity pulse counter register ... 361
15.2.14 QEIn_VLC: QEI n velocity capture register ... 362
15.2.15 QEIn_VLCOM: QEI n velocity compare register ... 362
15.2.16 QEIn_IER: QEI n interrupt enable register ... 363
15.2.17 QEIn_ISR: QEI n interrupt status register ... 364
15.2.18 QEIn_ISCR: QEI n interrupt status clear register ... 365
15.3 Functional description ... 366
15.3.1 Quadrature input signals ... 366
15.3.2 Position capture ... 368
15.3.3 Velocity capture ... 371
15.3.4 Velocity compare value ... 371
15.3.5 Setting examples ... 372
16 12-bit Analog-to-Digital Converter (ADC) ... 373
16.1 12-bit ADC block diagram ... 374
16.2 Internal channel wiring ... 375
16.3 Registers ... 376
16.3.1 ADCn_MR: ADC n mode register ... 377
16.3.2 ADCn_CSCR: ADC n current sequence/channel register ... 379
16.3.3 ADCn_CCR: ADC n clock control register ... 381
16.3.4 ADCn_TRG: ADC n trigger select register ... 382
16.3.5 ADCn_SCSR1: ADC n channel select 1 register ... 383

16.3.6 ADCn_SCSR2: ADC n channel select 2 register 384
16.3.7 ADCn_CR: ADC n control register 385
16.3.8 ADCn_SR: ADC n status register 386
16.3.9 ADCn_IER: ADC n interrupt enable register 387
16.3.10 ADCn_DDR: ADC n DMA data register 388
16.3.11 ADCn_DR: ADC n sequence 0 to 7 data register 389
16.3.12 ADCn_CMPR: ADC n channel compare register 390
16.3.13 ADCn_BCR: ADC n buffer control register 391
16.4 Functional description 392
16.4.1 ADC single mode timing diagram 392
16.4.2 ADC burst mode timing diagram 393
16.4.3 ADC sequential mode timing diagram 394
16.4.4 ADC multiple mode timing diagram 395
16.4.5 Setting examples 396
17 Analog Front End (AFE) 397
17.1 AFE block diagram 398
17.2 Registers 399
17.2.1 AFEn_CR: AFE n control register 400
17.2.2 CMP_DBR: comparator debounce register 401
17.2.3 CMP_ICR: comparator interrupt control register 402
17.2.4 CMP_SR: comparator status register 403
17.3 Functional description 404
17.3.1 Op-Amp configuration 404
17.3.2 Comparator reference and input voltage configuration 404
17.3.3 Comparator interrupt function 404
17.3.4 Setting examples 405
18 Cyclic Redundancy Check (CRC) 406
18.1 CRC block diagram 406
18.2 Registers 407
18.2.1 CRC_CTRL: CRC control register 408
18.2.2 CRC_INIT: CRC initial value register 409
18.2.3 CRC_IDR: CRC input data register 409
18.2.4 CRC_ODR: CRC output data register 410
18.2.5 CRC_STAT: CRC status register 410
18.3 Functional description 411
18.3.1 CRC polynomial structure 411
18.3.2 Setting examples 411
19 Electrical characteristics 412
19.1 Absolute maximum ratings 412
19.2 Recommended operating conditions 412
19.3 ADC characteristics 413
19.4 Power on Reset characteristics 413
19.5 Low voltage reset characteristics 414
19.6 Low voltage indicator characteristics 414
19.7 Analog front end characteristics 415

19.8 High frequency internal RC oscillator characteristics 416
19.9 Low frequency internal RC oscillator characteristics 416
19.10 DC electrical characteristics 417
19.11 Supply current characteristics 418
19.12 Internal Flash ROM characteristics 420
19.13 Main oscillator characteristics 421
19.14 PLL electrical characteristics 423
20 Package information 424
20.1 64 LQFP package information 424
20.2 48 LQFP package information 426
20.3 44 LQFP package information 427
21 Ordering information 428
22 Development tools 429
22.1 Compiler 429
22.2 Debugger 429
22.3 Programmer 430
22.3.1 E-PGM+ 430
22.3.2 Gang programmer 430
22.4 SWD debug mode and E-PGM+ connection 431
Revision history 432

List of figures

Figure 1. A33M11x Concept Block Diagram 1
Figure 2. A33M11x Block Diagram 24
Figure 3. LQFP 64 Pinouts 25
Figure 4. LQFP 48 Pinouts 26
Figure 4. LQFP 44 Pinouts 27
Figure 5. Interrupt Block Diagram 36
Figure 6. Memory Map 38
Figure 7. Connection Diagram of UART0 Boot 40
Figure 8. Connection Diagram of E-PGM+ and SWD Port 40
Figure 9. SCU Block Diagram 41
Figure 10. Clock Tree Configuration 42
Figure 11. Configuration of Miscellaneous Clocks 44
Figure 12. Clock Change Procedure 46
Figure 13. Peripheral Clock Select: n = 1, 2, 3, 4, 5, 6 and 7 48
Figure 14. Power-up Procedure 51
Figure 15. Warm Reset Diagram 52
Figure 16. LVR Reset Timing Diagram 52
Figure 17. Reset Tree Configuration 53
Figure 18. Transition between Operation Modes 54
Figure 19. SLEEP Mode Operation Sequence 55
Figure 20. STOP Mode Operation Sequence 57
Figure 21. LVI Block Diagram 85
Figure 22. LVR Block Diagram 87
Figure 23. PLL Block Diagram 97
Figure 24. HSE Operation Workflow 98
Figure 25. Clock Monitoring Workflow 99
Figure 26. PCU Block Diagram 103
Figure 27. GPIO Block Diagram 103
Figure 28. I/O Port Block Diagram: GPIO Pins 104
Figure 29. I/O Port Block Diagram: ADC and External Oscillator Pins 105
Figure 30. Port Function Block Diagram 121
Figure 31. Debouncing Logic 122
Figure 32. Port Debouncing Example 123
Figure 33. Port Function Block Diagram 123
Figure 34. Code Flash Memory Map: 256KB Code Flash 127
Figure 35. Data Flash Memory Map: 32KB Data Flash 128
Figure 36. Flash Program Sequence 153
Figure 37. Flash Burst Program Sequence 154
Figure 38. Flash Page Erase Sequence 155
Figure 39. Flash 2KB Sector Erase Sequence 156
Figure 40. Flash Chip Erase Sequence 157
Figure 41. DMAC Block Diagram 160
Figure 42. DMA Controller Block Diagram 165

Figure 43. Diagram of DMA Transfer Timing from a Peripheral to a Memory....................................166
Figure 44. Diagram of DMA Transfer Timing from a Memory to a Peripheral....................................167
Figure 45. N Times of DMA Transfers...167
Figure 46. WDT Block Diagram 170
Figure 47. Diagram of Interrupt Mode Operation Timing with External Clock as WDT Clock 176
Figure 48. 16-bit Timer Block Diagram 180
Figure 49. Basic Start and Match Operations...190
Figure 50. Normal Periodic Mode Operation 191
Figure 51. One-shot Mode Operation 193
Figure 52. PWM Output Operation 194
Figure 53. Timer Synchronization Example: When SSYNC = 1 195
Figure 54. Timer Synchronization Example: When CSYNC = 1 196
Figure 55. Timer Delayed Synchronization Example: When SSYNC = 1 197
Figure 56. Capture Mode Operation 198
Figure 57. ADC Triggering Timing Diagram: TRGMOD=0x00, 0x01 199
Figure 58. ADC Triggering Timing Diagram: TRGMOD=0x10 200
Figure 59. ADC Triggering Timing Diagram: TRGMOD=0x11 200
Figure 60. FRT Block Diagram...202
Figure 61. Match Interrupt Operation Timing Diagram 206
Figure 62. Overflow Interrupt Operation Timing Diagram 207
Figure 63. UART Block Diagram...210
Figure 64. Data Inversion Control Diagram 218
Figure 65. Sampling Timing of UART Receiver 224
Figure 66. Transmit Data Format Example..225
Figure 67. Transmit Interrupt Timing Diagram 225
Figure 68. Multi-sampling Timing of Start and Data Bits of Receive Data 226
Figure 69. Start Bit Detection and Single-/Multi-sampling Timing..227
Figure 70. Data Sampling Timing...228
Figure 71. Transmit Data Format Example..229
Figure 72. SPI Block Diagram...233
Figure 73. SPI Waveforms: STL, BTL, and SPL 240
Figure 74. SPI Transfer Timing 1/4: CPHA = 0, CPOL = 0, MSBF = 0 242
Figure 75. SPI Transfer Timing 2/4: CPHA = 0, CPOL = 1, MSBF = 1 242
Figure 76. SPI Transfer Timing 3/4: CPHA = 1, CPOL = 0, MSBF = 0 243
Figure 77. SPI Transfer Timing 4/4: CPHA = 1, CPOL = 1, MSBF = 1 243
Figure 78. DMA Handshake Workflow 244
Figure 79. I2C Block Diagram...250
Figure 80. INTDEL (Inter Delay) in Master Mode 254
Figure 81. SCL Low Timing..255
Figure 82. SCL High Timing 256
Figure 83. SDA Hold Timing...257
Figure 84. I2C Bus Bit Transfer: n = 0 and 1 258
Figure 85. START and STOP Conditions: n = 0 and 1 259
Figure 86. I2C Bus Data Transfer: n = 0 and 1 259
Figure 87. I2C Bus Response: n = 0 and 1...260

Figure 88. Clock Synchronization during Arbitration: n = 0 and 1 261
Figure 89. Arbitration Process between Two Masters: n = 0 and 1 262
Figure 90. Master Transmitter Flowchart 263
Figure 91. Master Receiver Flowchart 266
Figure 92. Slave Transmitter Flowchart 269
Figure 93. Slave Receiver Flowchart 271
Figure 94. MPWM Block Diagram: Normal Mode 276
Figure 95. MPWM Block Diagram: Individual Mode 277
Figure 96. MPWM Block Diagram: Phase U 278
Figure 97. Polarity Control Block 284
Figure 98. PWM Output Generation Chain 323
Figure 99. Up-count Mode Waveforms: MOTORB = 1, UPDOWN = 0, OLR.xLL =1, OLR.xHL =1 324
Figure 100. Up-count Mode Waveforms: MOTORB =1, UPDOWN =0, OLR.xLL =0, OLR.xHL =0 325
Figure 101. Up-down Count Mode Waveforms: MOTORB = 0, MCHMOD = 0, UPDOWN = 1 326
Figure 102. Two-channel Asymmetric Mode Waveforms: MOTORB = 0, MCHMOD = 00 327
Figure 103. One-channel Asymmetric Mode Waveforms: MOTORB = 0, MCHMOD = 01 328
Figure 104. One-channel Symmetric Mode Waveforms: MOTORB = 0, MCHMOD = 10 329
Figure 105. Two-channel Asymmetric Mode Waveforms: MOTORB = 3, MCHMOD = 00 330
Figure 106. One-channel Asymmetric Mode Waveforms: MOTORB = 3, MCHMOD = 01 331
Figure 107. One-channel Symmetric Mode Waveforms: MOTORB = 3, MCHMOD = 10 332
Figure 108. Dead Time Operation Timing Diagram: Symmetric Mode, DTMDSEL = 0 333
Figure 109. Dead Time Operation Timing Diagram: Asymmetric Mode, DTMDSEL = 0 334
Figure 110. Dead Time Polarity Settings 334
Figure 111. Dead Time Operation Timing Diagram: Symmetric Mode, DTMDSEL = 1 335
Figure 112. Dead Time Operation Timing Diagram: Asymmetric Mode, DTMDSEL = 1 335
Figure 113. Typical Dead Time Operation: TDUTY>TDT 336
Figure 114. H-side Minimum Pulse Timing: T_{DUTY}<T_{DT}<2xT_{DUTY} 337
Figure 115. H-side Zero Pulse Timing: T_{DT}>2xT_{DUTY} 338
Figure 116. L-side Minimum Pulse Timing: T_{DT}<Period-T_{DUTY} 339
Figure 117. L-side Zero Pulse Timing: T_{DT}>Period-T_{DUTY} 340
Figure 118. H-side Always On: T_{DUTY} = Period, Dead Time Set Disabled 341
Figure 119. L-side Always On: T_{DUTY} = 0, Dead Time Set Disabled 342
Figure 120. Symmetric PWM Timing 343
Figure 121. Asymmetric PWM Timing and Limitations in Detection 344
Figure 122. Protection and Overvoltage Block Diagram 345
Figure 123. ADC Trigger Timing Diagram 346
Figure 124. Example Timing of ADC Data Acquisition Triggered by an MPWM Event 347
Figure 125. Interrupt Generation Timing 348
Figure 126. QEI Block Diagram 353
Figure 127. State Changes in the Encoder 366
Figure 128. Capture Mode Edge Configuration: X2 and X4 Modes 368
Figure 129. Position Counter Reset: QEI_MAX Register Used 369
Figure 130. Position Counter Reset: Index Pulse Used 370
Figure 131. 12-bit ADC Block Diagram 374
Figure 132. 12-bit ADC Internal Channel Wiring 375

Figure 133. ADC Single Mode Timing: When ADCn.MR.AMOD = 0 392
Figure 134. ADC Burst Mode Timing: When ADCn.MR.ADMOD = 1 393
Figure 135. ADC Trigger Timing in Burst Mode: SEQCNT = 3’b111, 8 Sequential Conversion 393
Figure 136. ADC Sequential Mode Timing: When MR.AMOD = 0 and MR.SEQCNT ≠ 0 394
Figure 137. ADC Trigger Timing in Sequential Mode: SEQCNT = 3’b111, 8 Sequential Conversion . 394
Figure 138. ADC Multiple Mode Timing: When MR.AMOD = 2 and MR.SEQCNT ≠ 0 395
Figure 139. ADC Trigger Timing in Multiple Mode: SEQCNT = 3’b111, 8 Multiple Conversion 395
Figure 140. AFE Block Diagram 398
Figure 141. CRC Block Diagram 406
Figure 142. CRC Polynomial Structure 411
Figure 143. Crystal/Ceramic Oscillator 422
Figure 144. External Clock 422
Figure 145. Clock Timing Measurement at XIN 422
Figure 146. 64 LQFP (10x10) Package Outline 424
Figure 147. 64 LQFP (12x12) Package Outline 425
Figure 148. 48 LQFP (07x07) Package Outline 426
Figure 148. 44 LQFP (10x10) Package Outline 427
Figure 149. A33M11x Device Numbering Nomenclature 428
Figure 150. A-Link and Pin Descriptions 429
Figure 151. E-PGM+ (Single Writer) and Pin Descriptions 430
Figure 152. E-Gang4 and E-Gang6: for Mass Production 430
Figure 153. Connection between A33M11x Series and E-PGM+ using SWD Debugger Interface 431

List of tables

Table 1. A33M11x Series Features and Peripheral Counts 19
Table 2. Pin Description 28
Table 3. Interrupt Vector Map 33
Table 4. A33M11x Memory Boundary Addresses 37
Table 5. Boot Mode Pin List 39
Table 6. SCU Pins 41
Table 7. Clock Sources 43
Table 8. Flash Wait Control Recommendation 47
Table 9. Peripheral Clock Select 49
Table 10. Reset Sources of Cold Reset and Warm Reset 50
Table 11. Operation Mode 54
Table 12. STOP Mode Configuration 56
Table 13. Configurable Clocks in Each Operating Mode 58
Table 14. Base Address of SCU: Chip Configuration 59
Table 15. SCU Register Map: Chip Configuration 59
Table 16. Base Address of SCU 59
Table 17. SCU Register Map 59
Table 18. External Oscillator Control Settings 88
Table 19. PCU and GPIO Pins 102
Table 20. GPIO Alternative Function 106
Table 21. Base Address of PCU 108
Table 22. PCU and GPIO Register Map 108
Table 23. GPIO Block’s Interrupt Sources and Corresponding Pins 124
Table 24. Code Flash Memory Controller Features 127
Table 25. Data Flash Memory Controller Features 128
Table 26. Base Address of Code Flash Memory Controller 129
Table 27. CFMC Register Map 129
Table 28. Internal Flash Access Time by Operating Clock 131
Table 29. Available Operating Modes by Protection Level 138
Table 30. Base Address of Data Flash Memory Controller 143
Table 31. DFMC Register Map 143
Table 32. Internal Flash Access Time by Operating Clock 144
Table 33. Base Address of DMAC 161
Table 34. DMAC Register Map 161
Table 35. DMAC PERISEL Selection 162
Table 36. Base Address of WDT 171
Table 37. WDT Register Map 171
Table 38. Prescaled WDT Counter Clock Frequencies 177
Table 39. Pin Assignment of 16-bit Timer: External Pins 179
Table 40. Base Address of 16-bit Timer 181
Table 41. TIMER Register Map 181
Table 42. Base Address of FRT Interface 203
Table 43. FRT Register Map 203

Table 44. Pin Assignment of UART: External Pins 209
Table 45. Base Address of UART 211
Table 46. UART Register Map 211
Table 47. Interrupt IDs and Control 215
Table 48. Various Configurations to Create a Parity Bit 217
Table 49. Examples of Baud Rate Calculation 221
Table 50. Examples of Baud Rate Calculation Using a BFR Value 222
Table 51. Pin Assignment of SPI: External Pins 232
Table 52. Base Address of SPI 234
Table 53. SPI Register Map 234
Table 54. Pin Assignment of I2C: External Pins 249
Table 55. Base Address of I2C Interface 251
Table 56. I2C Register Map 251
Table 57. Pin Assignment of MPWM: External Pins 275
Table 58. Base Address of MPWM Interface 279
Table 59. MPWM Register Map 279
Table 60. MPWM Basic Output Level: MPWMn_OLR = 0x00 284
Table 61. Pin Assignment of QEI: External Pins 352
Table 62. Base Address of QEI 354
Table 63. QEI Register Map 354
Table 64. Encoder States 367
Table 65. Encoder State Changes 367
Table 66. Encoder Direction Changes 367
Table 67. Pin Assignment of ADC: External Pins 373
Table 68. Base Address of ADC 376
Table 69. ADC Register Map 376
Table 70. Trigger Source Table 382
Table 71. Pin Assignment of AFE: External Pins 397
Table 72. Base Address of AFE 399
Table 73. AFE Register Map 399
Table 74. Base Address of CRC 407
Table 75. CRC Register Map 407
Table 76. Absolute Maximum Ratings 412
Table 77. Recommended Operating Condition 412
Table 78. ADC Electrical Characteristics 413
Table 79. POR Electrical Characteristics 413
Table 80. Low Voltage Reset Characteristics 414
Table 81. Low Voltage Indicator Characteristics 414
Table 82. Op-amp Characteristics 415
Table 83. Comparator Characteristics 415
Table 84. High Frequency Internal RC Oscillator Characteristics 416
Table 85. Low Frequency (500KHz) Internal RC Oscillator Characteristics 416
Table 86. DC Electrical Characteristics 417
Table 87. Supply Current Characteristics: Normal and Sleep Mode 418
Table 88. Supply Current Characteristics: Stop Mode 419

Table 89. Internal Flash ROM Characteristics 420
Table 90. Main Oscillator Characteristics...421
Table 91. PLL Electrical Characteristics...423
Table 92. A33M11x Series Device Ordering Information 428

1 Description

A33M11x series is a 32-bit high-performance microcontroller with up to 256Kbytes of Flash memory. It is a powerful microcontroller which provides effective solutions to various electrical appliances which require both low power consumption and high performance.

1.1 Device overview

In this section, features of A33M11x series and peripheral counts are introduced.

Table 1. A33M11x Series Features and Peripheral Counts

Peripherals		Description
Core	CPU	• Maximum operating frequency: 96MHz • 32-bit ARM Cortex-M3 CPU • CPU register set: — Uses general-purpose registers specified by the 32-bit Thumb®-2 instruction set — Main stack pointer (MSP) and process stack pointer (PSP): R13 — Link register (LR): R14 — Program counter (PC): R15 • Data ordering format: Little-Endian • Harvard Architecture • AHB/APB
	Interrupt	• NVIC (Nested-Vectored Interrupt Controller) • Up to 86 peripheral interrupts supported. • Assignable with 16 different priority levels

Table 1. A33M11x Series Features and Peripheral Counts (continued)

Peripherals		Description
Memory	Code Flash	• Capacity : — A33M116: 256KB code Flash memory — A33M114: 128KB code Flash memory • A high-capacity code Flash memory built in • Max 28MHz Flash access speed • 512-B, 2-KB erases • Bulk erase • Read protection • Self-programming • CRC code generation and verification for the Flash memory • Endurance: 10,000 cycles • Lifetime: 10 years
	Data Flash	• Capacity: 32 KB • Max 28MHz access speed • 512-B, 2-KB erases • CRC code generation and verification for the Flash memory • Endurance: 100,000 Cycle • Lifetime: 10 years
	BOOT ROM	• Executes the processor's boot mode when receiving an input at the boot pin from an external circuit • UART boot modes • In-system programming — A user can program data into the internal Flash memory by setting an application board.
	SRAM	• Capacity: 16 KB • Usable as a program's work area • High-speed execution enables the execution of time-critical codes • Part of the SRAM can be remapped into an interrupt vector area
	Endurance	• 10,000 times at room temperature • Retention for 10 years
System Control Unit (SCU)	Operating frequency	• Up to 96Mhz
	Clock	• High speed internal oscillator (HSI) — 32MHz (±1.5% @-40℃ to +105℃) • Low speed internal oscillator (LSI) — 500KHz (±30% @-40℃ to +105℃) • External main oscillator (HSE): 4MHz to 16MHz • Phase-locked loop (PLL) frequency generator generates a high-speed clock (up to 96MHz)
System Control Unit (SCU)	Clock monitoring	• System Fail-Safe function by Clock Monitoring — External main oscillator(HSE) — Main system clock (MCLK)
	Operating mode	• RUN mode • SLEEP mode • STOP mode
	Reset	• nRESET pin reset • Core reset • Software reset

Table 1. A33M11x Series Features and Peripheral Counts (continued)

Peripherals		Description
		• POR (Power On Reset) • LVR (Low Voltage Reset) • WDTR (Watch Dog Timer Reset) • Reset due to clock oscillating error
	LDO	• Low-dropout (LDO) regulator built in for low-voltage operation
	POR	• The POR generator detects an internal VDC voltage and generates a reset signal
	LVI	• Supports interrupts • Supports wake-up from sleep mode
	Wake-up	• Wake-up by a general-purpose input/output (GPIO) pin • Wake-up by a free-run timer (FRT) • Wake-up by a watchdog timer (WDT) • Wake-up by a low-voltage indicator (LVI) • Wake-up by a Systick timer (SYSTICK)
General Purpose I/O (GPIO)		• Input/output (I/O) port for general purposes • LQFP-64 — I/O pins: 56 • LQFP-48 — I/O pins: 45 • LQFP-44 — I/O pins: 41 • Each pin can be set for one of the following modes: — Push-pull output — Open drain output — Input • The use of each pin can be set by setting the mux • Each pin can be configured as an external interrupt source, either the high-/low-level interrupt or the rising-/falling-edge interrupt • Pull-up/pull-down/debouncing can be set for each pin • Drive strength can be adjusted for each port pin • Each pin bit can be individually set/reset • Wake-up events triggered by external asynchronous inputs
Direct Memory Access Controller (DMA)		• 8-ch direct memory access (DMA) support peripherals • 8-/16-/32-bit data transfers • Compatible with 15 different types of peripherals — SPI0, SPI1, UART0, UART1, UART2, UART3, CRC, ADC0, ADC1
TIMER	16-bit Timer	• General-purpose 16-bit up-count timer • 8 channels — 8 timer n capture port (TnIO) input channels — 8 timer n output port (TnIO) output channels • Timer operating modes — Periodic timer mode — One-shot mode — PWM mode — Capture mode • Interrupt events — Timer/counter match interrupt

Table 1. A33M11x Series Features and Peripheral Counts (continued)

Peripherals		Description
		— Timer overflow interrupt • Input clock selection — Clocks are freely selectable through the miscellaneous clock control registers (MCCRs) — External clocks are selectable • Timer signals can be generated through TnIO pins • 10-bit prescaler
	WDT	• 32-bit down-count timer • Reset and periodic interrupts • Clocks are freely selectable through the miscellaneous clock control registers (MCCRs) • Eight different prescalers are selectable
	FRT	• 32-bit free-run timer — Capable of calculating the internal system time — 32-bit up-count timer • Interrupt events — Period interrupt — Overflow interrupt
Serial Interface	UART	• A total of four 16450 asynchronous serial communication ports • Configurable standard asynchronous communication bits (start, stop, and parity) • Flexible communication available through programming — 5- to 8-bit data transfers — Even-/odd-/non-parity generation and checking — 1-, 1.5-, or 2-stop bit generation and checking • 8-bit fraction controller and 16-bit baud rate generator
	SPI	• Two synchronous serial communication port channels • Master/slave operation • Loop-back mode • Programmable and flexible communication — 8-/9-/16-/17-bit data transmit/receive — SPI clock speed — Both least significant bit (LSB)-first and most significant bit (MSB)-first modes available
	I2C	• Standard I2C communication protocol • Two channels supported • Master and slave modes supported for each channel • 7-bit addressing supported for slave mode • SCL signal’s high/low periods and SDA signal’s hold time settable
Motor Pulse-Width Modulation	MPWM	• Two MPWM generators • Six channels (high and low signals of phases U, V, and W) generate different waveforms • 16-bit up-/down-counters • Six ADC trigger sources • Interrupt events — Bottom interrupts — Top (period) interrupts • Interval interrupt mode

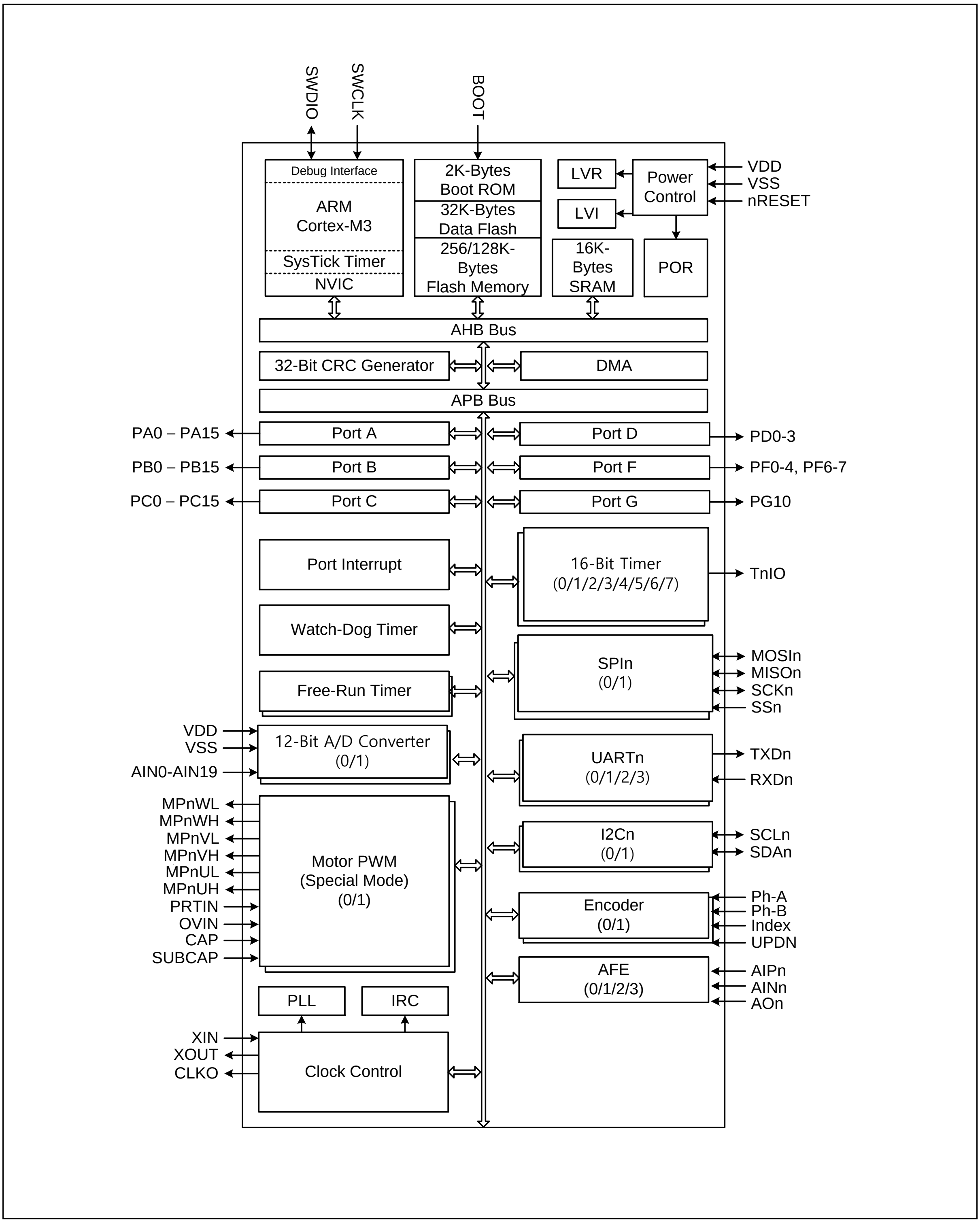
Table 1. A33M11x Series Features and Peripheral Counts (continued)

Peripherals		Description
		• Falling/rising dead time applicable • A special operating mode: — Phases U, V, and W are independently controlled — Different carrier counters running for phases U, V, and W — Different duties and periods configurable for phases U, V, and W — Different interrupts used for phases U, V, and W — Capture functionality • Protection and over-voltage detection supported
Quadrature Encoder Interface	QEI	• Two QEI channels • Three input pins for two phase signals and an index pulse • Programmable noise input filters • Displays counter pulses and counter direction • 32-bit up-/down-counters • Velocity capture using a timer
12-bit A/D Converter	ADC	• Two independent ADC blocks • 22 analog input channels • A number of operating modes: — Single conversion — Sequence conversion — Burst conversion — Multiple conversion • Up to eight sequential conversions supported • Software triggers supported • Three internal trigger sources (MPWM and timers) supported • Sample time and hold time are adjustable
Analog Front End	AFE	• 4 channels of OPAMP that can operate independently of the comparator • Available to connect with ADC
Cyclic Redundancy Check	CRC	• CRC operating modes: — CRC32 (0x04C1_1DB7) — CRC16 (0x8005) — CRC8 (0x07) — CRC7 (0x09) • Input/output data reversion supported • Compatible with DMA
Operating Voltage		• 2.5V to 5.5V
Operating temperature		• Commercial grade (-40℃ to +105℃)
Package		• Three types of package options — LQFP-64 — LQFP-48 — LQFP-44

1.2 A33M11x block diagram

In this section, the A33M11x series with peripherals is described in block diagram.

Figure 2. A33M11x Block Diagram



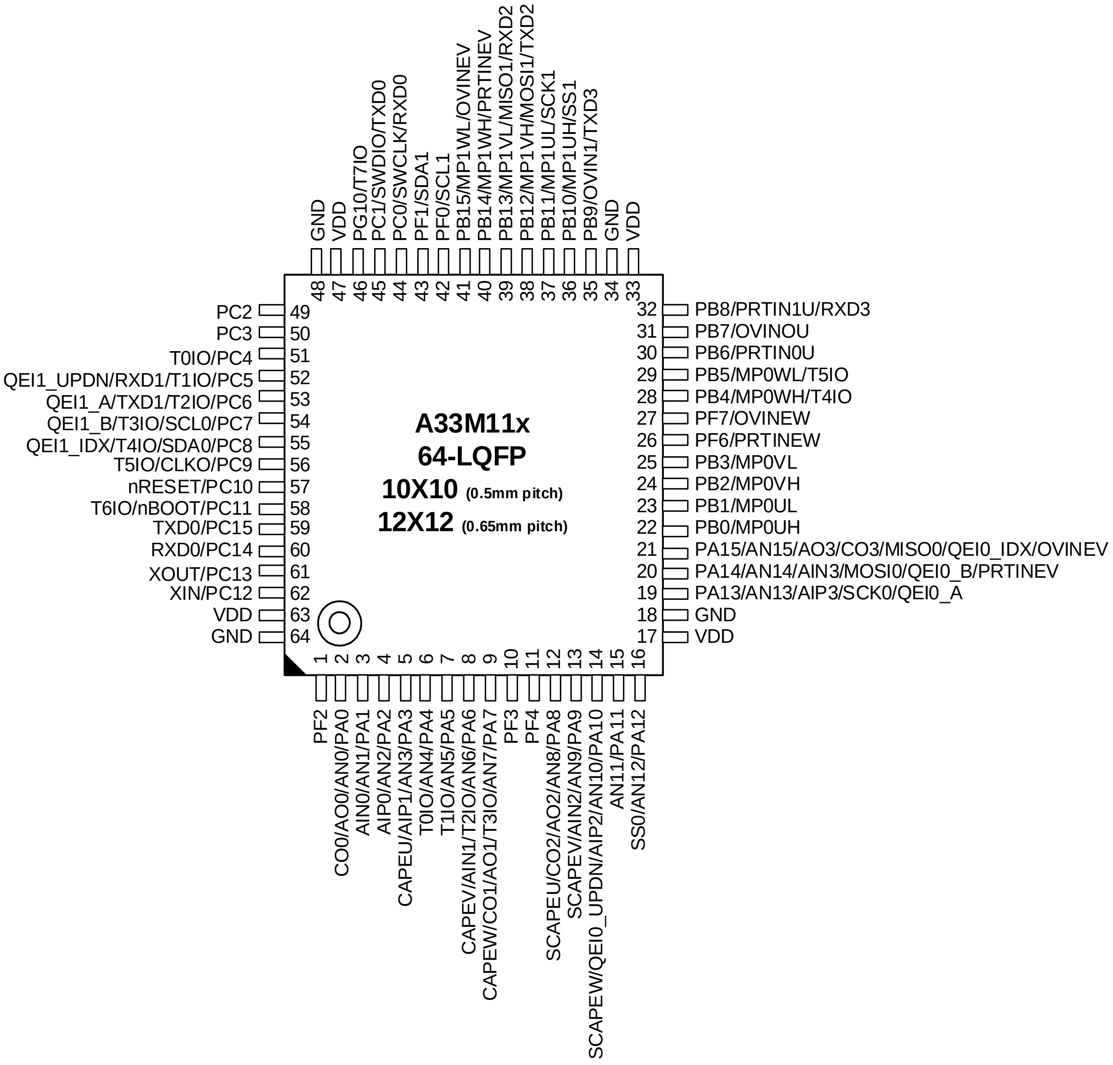
2 Pinouts and pin descriptions

In this chapter, pinouts and pin descriptions of A33M11x series are introduced.

2.1 Pinouts

2.1.1 A33M116RL/A33M116RM/A33M114RL (LQFP-64)

Figure 3. LQFP 64 Pinouts

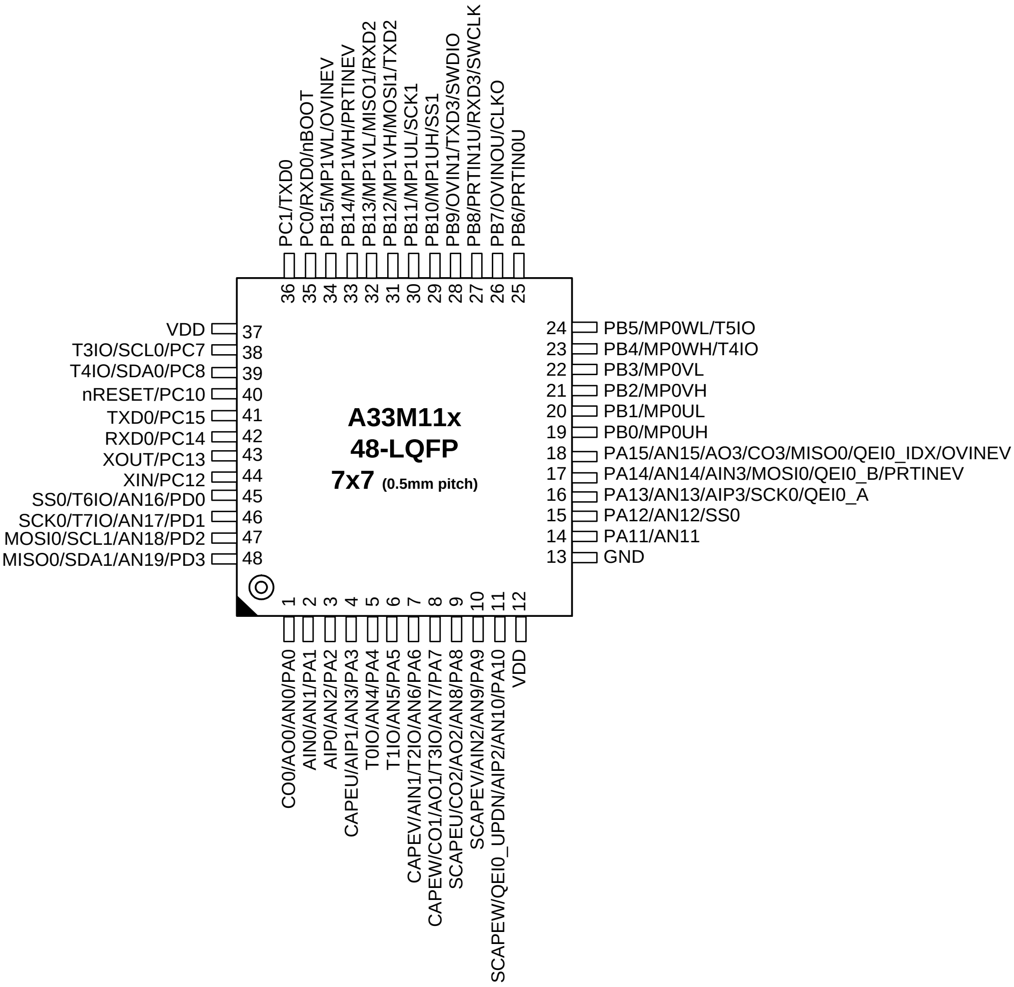


NOTES:

1. PC[8:7] are Open-drain pins. When using a port corresponding to these pins for Timer, I2C, QEI, etc., applications must use external Pull-up resistors.
2. Since the 64-pin package, the 48-pin package and the 44-pin package of the A33M11x series have different Boot pins and SWD pins (SWDIO, SWCLK), users need to be careful to set ports.
   - 64-pin package: PC0(PC_MR1.P0MUX:0b011/SWCLK), PC1(PC_MR1.P1MUX:0b011/SWDIO), PC11(PC_MR2.P11MUX:0b011/nBOOT)
   - 48-pin and 44-pin package: PB8(PB_MR2.P8MUX:0b100/SWCLK), PB9(PB_MR2.P9MUX:0b100/SWDIO), PC0(PC_MR1.P0MUX:0b100/nBOOT)

2.1.2 A33M116CL/A33M114CL (LQFP-48)

Figure 4. LQFP 48 Pinouts

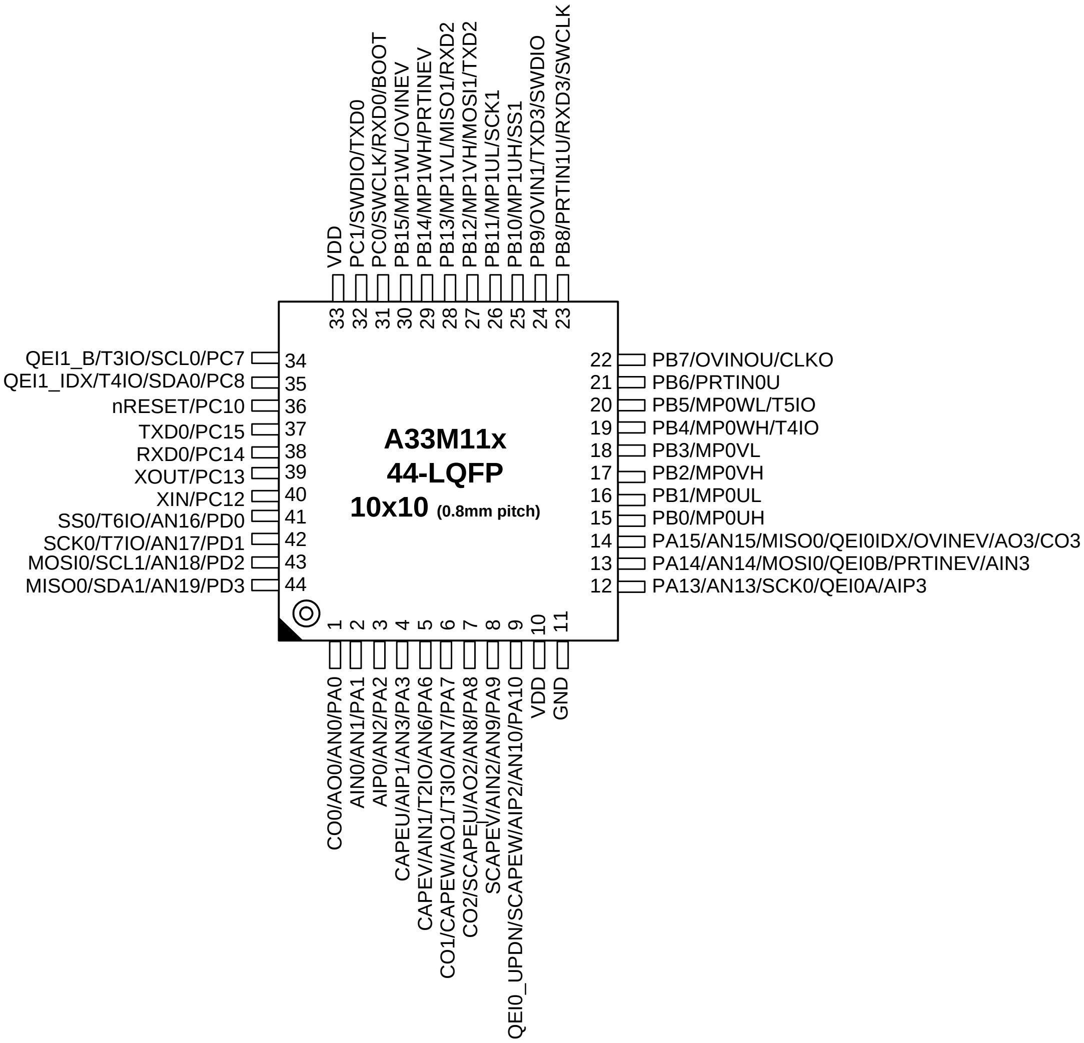


NOTES:

1. PC[8:7] are Open-drain pins. When using a port corresponding to these pins for Timer, I2C, QEI, etc., applications must use external Pull-up resistors.
2. Since the 64-pin package, the 48-pin package and the 44-pin package of the A33M11x series have different Boot pins and SWD pins (SWDIO, SWCLK), users need to be careful to set ports.
   - 64-pin package: PC0(PC_MR1.P0MUX:0b011/SWCLK), PC1(PC_MR1.P1MUX:0b011/SWDIO), PC11(PC_MR2.P11MUX:0b011/nBOOT)
   - 48-pin and 44-pin package: PB8(PB_MR2.P8MUX:0b100/SWCLK), PB9(PB_MR2.P9MUX:0b100/SWDIO), PC0(PC_MR1.P0MUX:0b100/nBOOT)

2.1.3 A33M114SN (LQFP-44)

Figure 5. LQFP 44 Pinouts



NOTES:

1. PC[8:7] are Open-drain pins. When using a port corresponding to these pins for Timer, I2C, QEI, etc., applications must use external Pull-up resistors.
2. Since the 64-pin package, the 48-pin and the 44-pin package of the A33M11x series have different Boot pins and SWD pins (SWDIO, SWCLK), users need to be careful to set ports.
   - 64-pin package: PC0(PC_MR1.P0MUX:0b011/SWCLK), PC1(PC_MR1.P1MUX:0b011/SWDIO), PC11(PC_MR2.P11MUX:0b011/nBOOT)
   - 48-pin and 44-pin package: PB8(PB_MR2.P8MUX:0b100/SWCLK), PB9(PB_MR2.P9MUX:0b100/SWDIO), PC0(PC_MR1.P0MUX:0b100/nBOOT)

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 five 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	44-pin				
1	-	-	PF2*	IOUDS	PORT F Bit 2 Input/Output	
2	1	1	PA0*	IOUDS	PORT A Bit 0 Input/Output	
			AN0	IA	Analog Input 0	
			AO0	AO	OPAMP0 Output	
			CO0	AO	Comparator0 Output	
3	2	2	PA1*	IOUDS	PORT A Bit 1 Input/Output	
			AN1	IA	Analog Input 1	
			AIN0	IO	OPAMP0 & Comparator0 Analog Input (-)	
4	3	3	PA2*	IOUDS	PORT A Bit 2 Input/Output	
			AN2	IA	Analog Input 2	
			AIP0	IO	OPAMP0 & Comparator0 Analog Input (+)	
5	4	4	PA3*	IOUDS	PORT A Bit 3 Input/Output	
			AN3	IA	Analog Input 3	
			AIP1	IO	OPAMP1 & Comparator1 Analog Input (+)	
			CAPEU	I	Extend PWM Capture U phase	
6	5	-	PA4*	IOUDS	PORT A Bit 4 Input/Output	
			AN4	IA	Analog Input 4	
			T0IO	IO	Timer 0 Input/Output	
7	6	-	PA5*	IOUDS	PORT A Bit 5 Input/Output	
			AN5	IA	Analog Input 5	
			T1IO	IO	Timer 1 Input/Output	
8	7	5	PA6*	IOUDS	PORT A Bit 6 Input/Output	
			AN6	IA	Analog Input 6	
			T2IO	IO	Timer 2 Input/Output	
			AIN1	IO	OPAMP1 & Comparator1 Analog Input (-)	
			CAPEV	I	Extend PWM Capture V phase	
9	8	6	PA7*	IOUDS	PORT A Bit 7 Input/Output	
			AN7	IA	Analog Input 7	
			T3IO	IO	Timer 3 Input/Output	
			AO1	AO	OPAMP1 Output	
			CO1	AO	Comparator1 Output	
			CAPEW	I	Extend PWM Capture W phase	
10	-	-	PF3*	IOUDS	PORT F Bit 3 Input/Output	
11	-	-	PF4*	IOUDS	PORT F Bit 4 Input/Output	

Table 2. Pin Description (continued)

Pin no.			**Pin name**	**Type**	**Description**	**Remark**
64-pin	**48-pin**	44-pin				
12	9	7	PA8*	IOUDS	PORT A Bit 8 Input/Output	
			AN8	IA	Analog Input 8	
			AO2	AO	OPAMP2 Output	
			CO2	AO	Comparator2 Output	
			SCAPEU	I	Extend PWM Sub Capture U phase	
13	10	8	PA9*	IOUDS	PORT A Bit 9 Input/Output	
			AN9	IA	Analog Input 9	
			AIN2	IO	OPAMP2 & Comparator 2 Analog Input (-)	
			SCAPEV	I	Extend PWM Sub Capture V phase	
14	11	9	PA10*	IOUDS	PORT A Bit 10 Input/Output	
			AN10	IA	Analog Input 10	
			AIP2	IO	OPAMP2 & Comparator2 Analog Input (+)	
			SCAPEU	I	Extend PWM Sub Capture U phase	
			QEI0_UDPN	O	QEI0 Output of Phase Direction	
15	14	-	PA11*	IOUDS	PORT F Bit 11 Input/Output	
			AN11	IA	Analog Input 11	
16	15	-	PA12*	IOUDS	PORT F Bit 12 Input/Output	
			AN12	IA	Analog Input 12	
			SS0	IO	SPI Channel 0 Select Signal Input/Output	
17	12	10	VDD	P	VDD	
18	13	11	GND	P	Ground	
19	16	12	PA13*	IOUDS	PORT A Bit 13 Input/Output	
			AN13	IA	Analog Input 13	
			SCK0	IO	SPI Channel 1 Clock Input/Output	
			QEI0_A	I	Input of QEI0 Phase A	
			AIP3	IO	OPAMP3 & Comparator3 Analog Input (+)	
20	17	13	PA14*	IOUDS	PORT A Bit 14 Input/Output	
			AN14	IA	Analog Input 14	
			MOSI0	IO	SPI Channel 0 Master Out/Slave In Signal	
			QEI0_B	I	Input of QEI0 Phase B	
			PRTINEV	I	Extend PWM Phase V Protection Input	
			AIN3	IO	OPAMP3 & Comparator3 Analog Input (-)	
21	18	14	PA15*	IOUDS	PORT A Bit 15 Input/Output	
			AN15	IA	Analog Input 15	
			MISO0	IO	SPI Channel 0 Master In/Slave Out Signal	
			QEI0_IDX	I	Input of QEI0 Index	
			OVINEV	I	Extend PWM Phase V Over-voltage Input	
			AO3	AO	OPAMP3 Output	
			CO3	AO	Comparator3 Output	
22	19	15	PB0*	IOUDS	PORT B Bit 0 Input/Output	
			MP0UH	O	PWM0 UH Output	
23	20	16	PB1*	IOUDS	PORT B Bit 1 Input/Output	
			MP0UL	O	PWM Channel 0 UL Output	
24	21	17	PB2*	IOUDS	PORT B Bit 2 Input/Output	
			MP0VH	O	PWM Channel 0 VH Output	
25	22	18	PB3*	IOUDS	PORT B Bit 3 Input/Output	
			MP0VL	O	PWM Channel 0 VL Output	
26	-	-	PF6*	IOUDS	PORT F Bit 6 Input/Output	
			PRTINEW	I	Extend PWM Phase W Protection Input	
27	-	-	PF7*	IOUDS	PORT F Bit 7 Input/Output	
			OVINEW	I	Extend PWM Phase W Over-voltage Input	
28	23	19	PB4*	IOUDS	PORT B Bit 4 Input/Output	
			MP0WH	Output	PWM Channel 0 WH Output	
			T4IO	IO	Timer 4 Input/Output	
29	24	20	PB5*	IOUDS	PORT B Bit 5 Input/Output	
			MP0WL	O	PWM Channel 0 WL Output	
			T5IO	IO	Timer 5 Input/Output	
30	25	21	PB6*	IOUDS	PORT B Bit 6 Input/Output	

Table 2. Pin Description (continued)

Pin no.			Pin name	Type	Description	Remark
64-pin	**48-pin**	44-pin				
			PRTIN0U	I	PWM0 Protection Input Signal Extend PWM 0 Phase U Protection Input	
31	26	22	PB7*	IOUDS	PORT B Bit 7 Input/Output	
			OVIN0U	I	PWM0 Over-voltage Input Signal Extend PWM 0 Phase U Over voltage Input	
			CLKO	O	System Clock Output	
32	27	23	PB8*	IOUDS	PORT B Bit 8 Input/Output	
			PRTIN1U	I	PWM1 Protection Input Signal Extend PWM 1 Phase U Protection Input	
			RXD3	I	UART Channel 3 RXD Input	
			SWCLK	I	SWD Clock Input	
33	-	-	VDD	P	VDD	
34	-	-	GND	P	Ground	
35	28	24	PB9*	IOUDS	PORT B Bit 9 Input/Output	
			OVIN1U	I	PWM1 Over-voltage Input Signal Extend PWM 1 Phase U Over voltage Input	
			TXD3	O	UART Channel 3 TXD Output	
			SWDIO	IO	SWD Data Input/Output	
36	29	25	PB10*	IOUDS	PORT B Bit 10 Input/Output	
			MP1UH	O	PWM Channel 1 UH Output	
			SS1	IO	SPI Channel 1 Select Signal Input/Output	
37	30	26	PB11*	IOUDS	PORT B Bit 11 Input/Output	
			MP1UL	O	PWM Channel 1 UL Output	
			SCK1	IO	SPI Channel 1 Clock Input/Output	
38	31	27	PB12*	IOUDS	PORT B Bit 12 Input/Output	
			MP1VH	O	PWM Channel 1 VH Output	
			MOSI1	IO	SPI Channel 1 Master Out/Slave In Signal	
			TXD2	O	UART Channel 2 TXD Output	
39	32	28	PB13*	IOUDS	PORT B Bit 13 Input/Output	
			MP1VL	O	PWM Channel 1 VL Output	
			MISO1	IO	SPI Channel 1 Master In/Slave Out Signal	
			RXD2	I	UART Channel 2 RXD Input	
40	33	29	PB14*	IOUDS	PORT B Bit 14 Input/Output	
			MP1WH	O	PWM Channel 1 WH Output	
			PRTINEV	I	Extend PWM Phase V Protection Input	
41	34	30	PB15*	IOUDS	PORT B Bit 15 Input/Output	
			MP1WL	O	PWM Channel 1 WL Output	
			OVINEV	I	Extend PWM Phase V Over-voltage Input	
42	-	-	PF0*	IOUDS	PORT F Bit 0 Input/Output	
			SCL1	O	I2C Channel 1 Output	
43	-	-	PF1*	IOUDS	PORT F Bit 1 Input/Output	
			SDA1	IO	I2C Channel 1 SDA Input/Output	
44	35	31	PC0	IOUDS	PORT C Bit 0 Input/Output	
			SWCLK*	I	SWD Clock Input	Pull-up
			RXD0	I	UART Channel 0 RXD Input	
			nBOOT	I	Boot Mode Selection Input	
45	36	32	PC1	IOUDS	PORT C Bit 1 Input/Output	
			SWDIO*	IO	SWD Data Input/Output	Pull-up
			TXD0	O	UART Channel 0 TXD Output	
46	-	-	PG10*	IOUDS	PORT G Bit 10 Input/Output	
			T7IO	IO	Timer 7 Input/Output	
47	37	33	VDD	P	VDD	
48	-	-	GND	P	Ground	
49	-	-	PC2*	IOUDS	PORT C Bit 2 Input/Output	
50	-	-	PC3*	IOUDS	PORT C Bit 3 Input/Output	
51	-	-	PC4*	IOUDS	PORT C Bit 4 Input/Output	
			T0IO	IO	Timer 0 Input/Output	
52	-	-	PC5*	IOUDS	PORT C Bit 5 Input/Output	

Table 2. Pin Description (continued)

Pin no.			**Pin name**	**Type**	**Description**	**Remark**
64-pin	**48-pin**	44-pin				
			T1IO	IO	Timer 1 Input/Output	
			RXD1	I	UART Channel 1 RXD Input	
			QEI1_UPDN	O	QEI1 Output of Phase Direction	
53	-	-	PC6*	IOUDS	PORT C Bit 6 Input/Output	
			T2IO	IO	Timer 2 Input/Output	
			TXD1	O	UART Channel 1 TXD Output	
			QEI1_A	I	Input of QEI1 Phase A	
54	38	34	PC7*	IOUDS	PORT C Bit 7 Input/Output	
			T3IO	IO	Timer 3 Input/Output	
			SCL0	O	I2C Channel 0 Output	Open-drain
			QEI1_B	I	Input of QEI1 Phase B	
55	39	35	PC8*	IOUDS	PORT C Bit 8 Input/Output	
			T4IO	IO	Timer 4 Input/Output	
			SDA0	IO	I2C Channel 0 SDA Input/Output	Open-drain
			QEI1_IDX	I	Input of QEI1 Index	
56	-	-	PC9*	IOUDS	PORT C Bit 9 Input/Output	
			T5IO	IO	Timer 5 Input/Output	
			CLKO	O	System Clock Output	
57	40	36	PC10	IOUDS	PORT C Bit 10 Input/Output	
			nRESET*	I	External Reset Input	Pull-up
58	-	-	PC11	IOUDS	PORT C Bit 11 Input/Output	
			T6IO	IO	Timer 6 Input/Output	
			nBOOT*	I	Boot Mode Selection Input	Pull-up
59	41	37	PC15*	IOUDS	PORT C Bit 15 Input/Output	
			TXD0	O	UART Channel 0 TXD Output	
60	42	38	PC14*	IOUDS	PORT C Bit 14 Input/Output	
			RXD0	I	UART Channel 0 RXD Input	
61	43	39	PC13*	IOUDS	PORT C Bit 13 Input/Output	
			XOUT	OA	External Crystal Oscillator Output	
62	44	40	PC12*	IOUDS	PORT C Bit 12 Input/Output	
			XIN	IA	External Crystal Oscillator Input	
-	45	41	PD0*	IOUDS	PORT D Bit 0 Input/Output	
			AN16	IA	Analog Input 16	
			T6IO	IO	Timer 6 Input/Output	
			SS0	IO	SPI Channel 0 Select Signal Input/Output	
-	46	42	PD1*	IOUDS	PORT D Bit 1 Input/Output	
			AN17	IA	Analog Input 17	
			T7IO	IO	Timer 7 Input/Output	
			SCK0	IO	SPI Channel 0 Clock Input/Output	
-	47	43	PD2*	IOUDS	PORT D Bit 2 Input/Output	
			AN18	IA	Analog Input 18	
			MOSI0	IO	SPI Channel 0 Mast Out/Slave Out Signal	
			SCL1	O	I2C Channel 0 Output	
-	48	44	PD3*	IOUDS	PORT D Bit 3 Input/Output	
			AN19	IA	Analog Input 19	
			MISO0	IO	SPI Channel 0 Master In/Slave Out Signal	
			SDA1	IO	I2C Channel 0 SDA Input/Output	
63	-		VDD	P	VDD	
64	-		GND	P	Ground	

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. * means ‘selected pin function after reset condition’, which is a 64-pin standard.(The initial value of the pin is different depending on the package type)
3. Pin order may be changed with revision notice.

4. PC11 (nBOOT), PC10 (nRESET), PC1 (SWDIO) and PC0 (SWCLK) are the default pull-up pins.
5. Do not configure unused pins as floating inputs.
6. After a reset, the internal pull-up for the boot pin is enabled.
7. After a reset, the internal pull-up for the serial wire clock (SWCLK) and the serial wire data I/O (SWDIO) is enabled.
8. The SWCLK and SWDIO pins should not be switched to other functions while they are being used.
9. PC7-8 pins are open-drain ports.
10. When a function other than a clock is set through the PC12 (XIN) and PC13 (XOUT) pins, other functions than the set clock will not operate normally if the clock is enabled by software.

3 System and memory overview

3.1 System architecture

Main system of A33M11x series consists of the followings:

- ARM® Cortex-M3 core
- General purpose DMA
- Internal SRAM, Flash memory
- Two AHB buses

3.1.1 Cortex-M3 core

The ARM® Cortex-M3 processor is a low-power processor that features low gate count, low interrupt latency, and low-cost debug. The processor is designed for embedded, high-level applications that require fast interrupt response, including microcontrollers, automatic and industrial control systems.

For detailed information on the Cortex-M3, refer to document DDI337 provided by the ARM.

3.1.2 Interrupt controller

Table 3. Interrupt Vector Map

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

Table 3. Interrupt Vector Map (continued)

Priority	Vector address	Interrupt source
0	0x0000_0040	LVI
1	0x0000_0044	SYSCLKFAIL
2	0x0000_0048	HSEFAIL
3	0x0000_004C	Reserved
4	0x0000_0050	
5	0x0000_0054	
6	0x0000_0058	WDT
7	0x0000_005C	Reserved
8	0x0000_0060	FRT0
9	0x0000_0064	Reserved
10	0x0000_0068	
11	0x0000_006C	CFMC
12	0x0000_0070	DFMC
13	0x0000_0074	Reserved
14	0x0000_0078	
15	0x0000_007C	TIMER0
16	0x0000_0080	TIMER1
17	0x0000_0084	TIMER2
18	0x0000_0088	TIMER3
19	0x0000_008C	TIMER4
20	0x0000_0090	TIMER5
21	0x0000_0094	TIMER6
22	0x0000_0098	TIMER7
23	0x0000_009C	Reserved
24	0x0000_00A0	
25	0x0000_00A4	
26	0x0000_00A8	
27	0x0000_00AC	
28	0x0000_00B0	
29	0x0000_00B4	
30	0x0000_00B8	
31	0x0000_00BC	QEI0
32	0x0000_00C0	QEI1
33	0x0000_00C4	Reserved
34	0x0000_00C8	
35	0x0000_00CC	
36	0x0000_00D0	GIPOA
37	0x0000_00D4	GPIOB
38	0x0000_00D8	GPIOC
39	0x0000_00DC	GPIOD
40	0x0000_00E0	Reserved
41	0x0000_00E4	GPIOF
42	0x0000_00E8	GPIOG
43	0x0000_00EC	Reserved
44	0x0000_00F0	
45	0x0000_00F4	MPWM0PROT
46	0x0000_00F8	MPWM0OVV

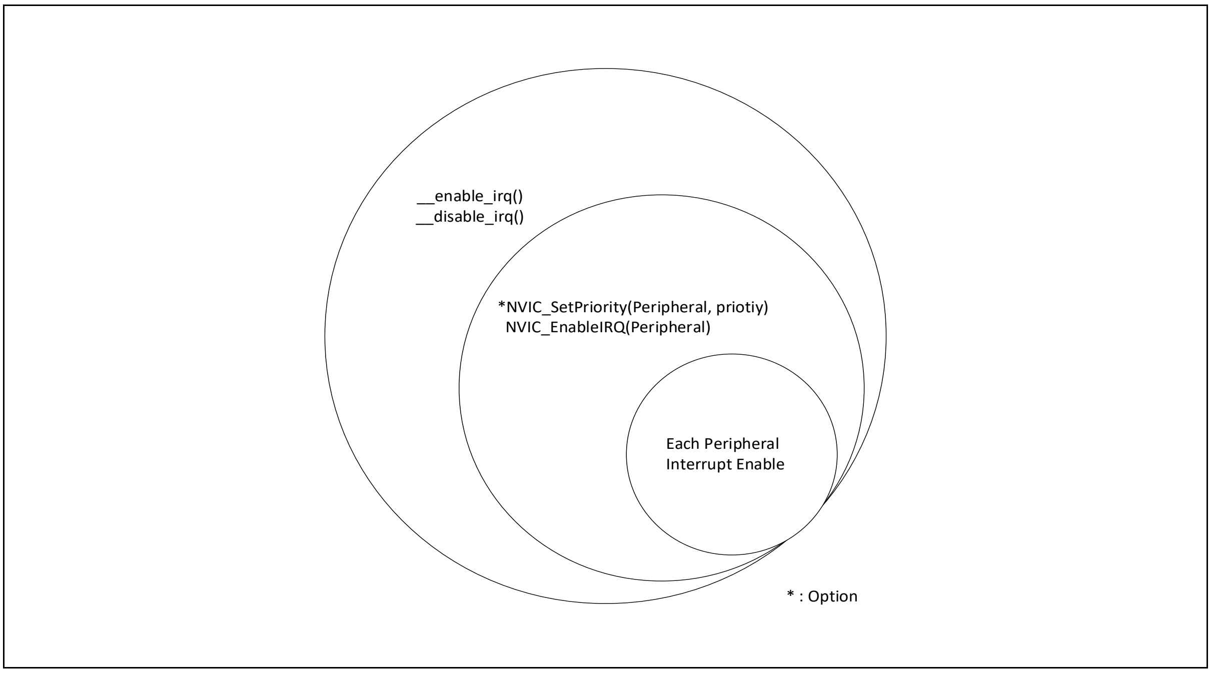
Table 3. Interrupt Vector Map (continued)

Priority	Vector address	Interrupt source
47	0x0000_00FC	MPWM0(U)
48	0x0000_0100	MPWM0(V)
49	0x0000_0104	MPWM0(W)
50	0x0000_0108	MPWM1PROT
51	0x0000_010C	MPWM1OVV
52	0x0000_0110	MPWM1(U)
53	0x0000_0114	MPWM1(V)
54	0x0000_0118	MPWM1(W)
55	0x0000_011C	SPI0
56	0x0000_0120	SPI1
57	0x0000_0124	Reserved
58	0x0000_0128	
59	0x0000_012C	
60	0x0000_0130	I2C0
61	0x0000_0134	I2C1
62	0x0000_0138	Reserved
63	0x0000_013C	UART0
64	0x0000_0140	UART1
65	0x0000_0144	UART2
66	0x0000_0148	UART3
67	0x0000_014C	Reserved
68	0x0000_0150	
69	0x0000_0154	
70	0x0000_0158	
71	0x0000_015C	
72	0x0000_0160	
73	0x0000_0164	
74	0x0000_0168	ADC0
75	0x0000_016C	ADC1
76	0x0000_0170	Reserved
77	0x0000_0174	
78	0x0000_0178	
79	0x0000_017C	AFE0
80	0x0000_0180	AFE1
81	0x0000_0184	AFE2
82	0x0000_0188	AFE3
83	0x0000_018C	Reserved
84	0x0000_0190	
85	0x0000_0194	CRC

NOTES:

1. Each Interrupt Priority Level Register occupies 1 byte (8 bits).NVIC registers in the Cortex-M3 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 = 4
2. Figure 6 shows inclusion relationship between peripheral interrupt enables and functions in core. An interrupt doesn’t work if only the peripheral interrupts are enabled. A user must enable the function in core to enable the interrupts.
   * __enable_irq > NVCI_EnableIRQ(Peripheral) > Each Peripheral Interrupt

Figure 6. 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 address assigned for each peripheral of the A33M11x series.

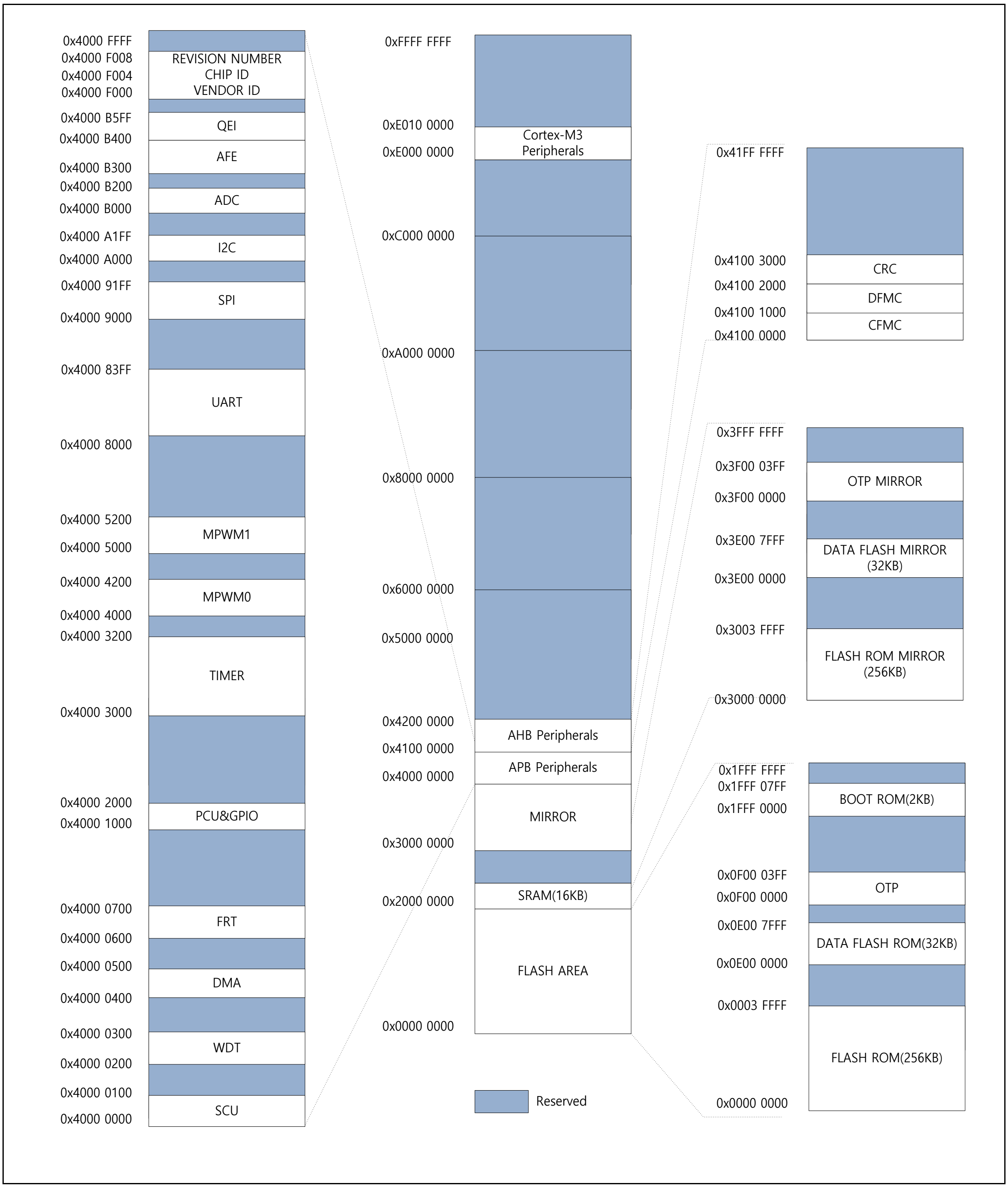
Table 4. A33M11x Memory Boundary Addresses

Boundary address	Memory area
0x4000_0000	SCU
0x4000_0200	WDT
0x4000_0400	DMA 0/1/2/3/4/5/6/7
0x4000_0600	FRT
0x4000_1000	PCU A/B/C/D/F/G
0x4000_3000	TIMER 0/1/2/3/4/5/6/7
0x4000_4000	MPWM0
0x4000_5000	MPWM1
0x4000_8000	UART 0/1/2/3
0x4000_9000	SPI 0/1
0x4000_A000	I2C 0/1
0x4000_B000	ADC 0/1
0x4000_B300	AFE 0/1/2/3
0x4000_B400	QEI 0/1
0x4100_0000	CFMC
0x4100_1000	DFMC
0x4100_2000	CRC
0x2000_0000	Internal SRAM

3.2.2 Memory map

Figure 7 shows addressable memory space in memory map.

Figure 7. Memory Map



3.2.3 Embedded SRAM

The A33M11x series has a block of 0-wait on-chip SRAM. Size of the SRAM is 16KB and its base address is 0x2000_0000.

This 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

The A33M11x series provides internal 256KB code Flash memory and a controller. This is enough to control the general system. Self-programming is available and ISP and SWD programming is also supported in boot mode or in 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 28MHz bus frequency.

3.2.5 Boot mode

3.2.5.1 Boot mode pins

The A33M11x 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:

- UART boot uses TXD0//RXD0 ports.

Pins for the boot mode are listed in Table 5.

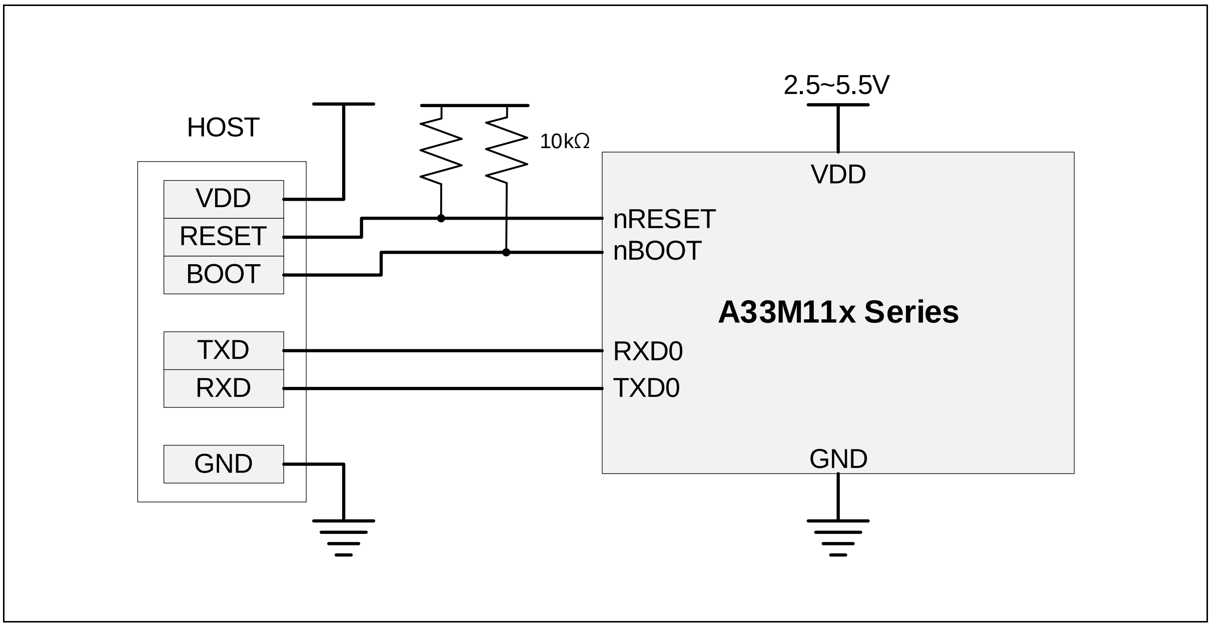
Table 5. Boot Mode Pin List

Block	Pin name	Dir	Description
SYSTEM	nRESET/PC10	I	Reset input signal
	nBOOT/PC11	I	Boot mode setting pin
UART0	RXD0/PC14	I	UART boot receive data
	TXD0/PC15	O	UART boot transmit data

3.2.5.2 Boot mode connections

Users can design a target board using any of boot mode ports such as UART mode of UART0. Sample connection diagrams of boot mode are introduced in the following figures:

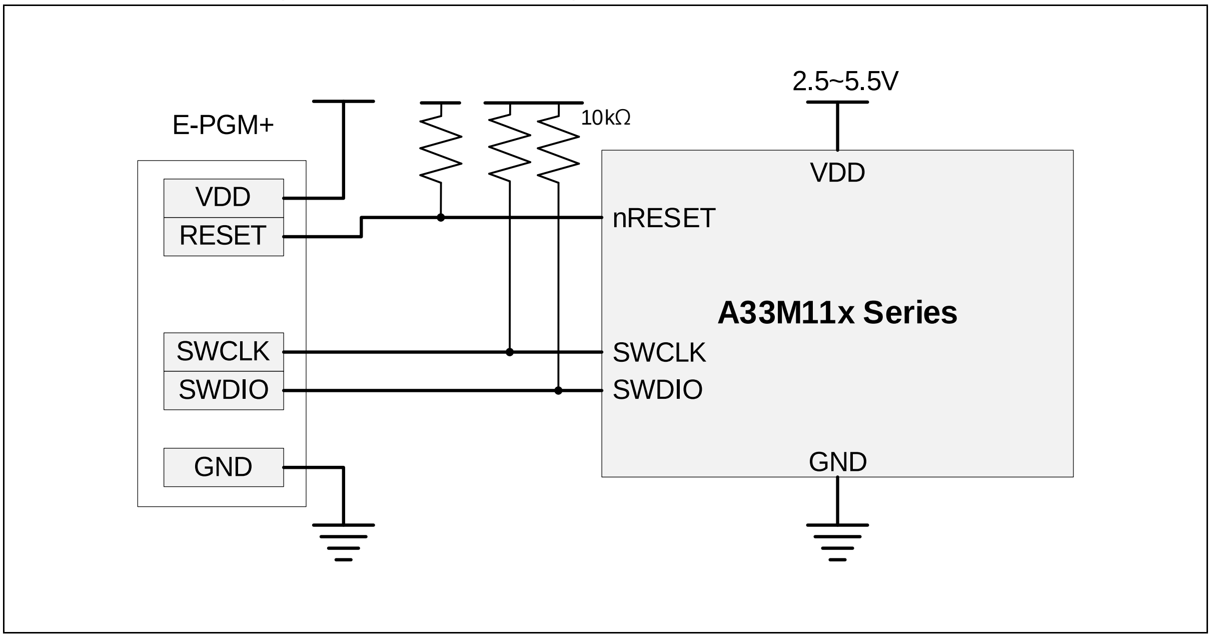
Figure 8. Connection Diagram of UART0 Boot



3.2.5.3 SWD mode connections

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

Figure 9. Connection Diagram of E-PGM+ and SWD Port



4 System Control Unit (SCU)

A33M11x series has 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

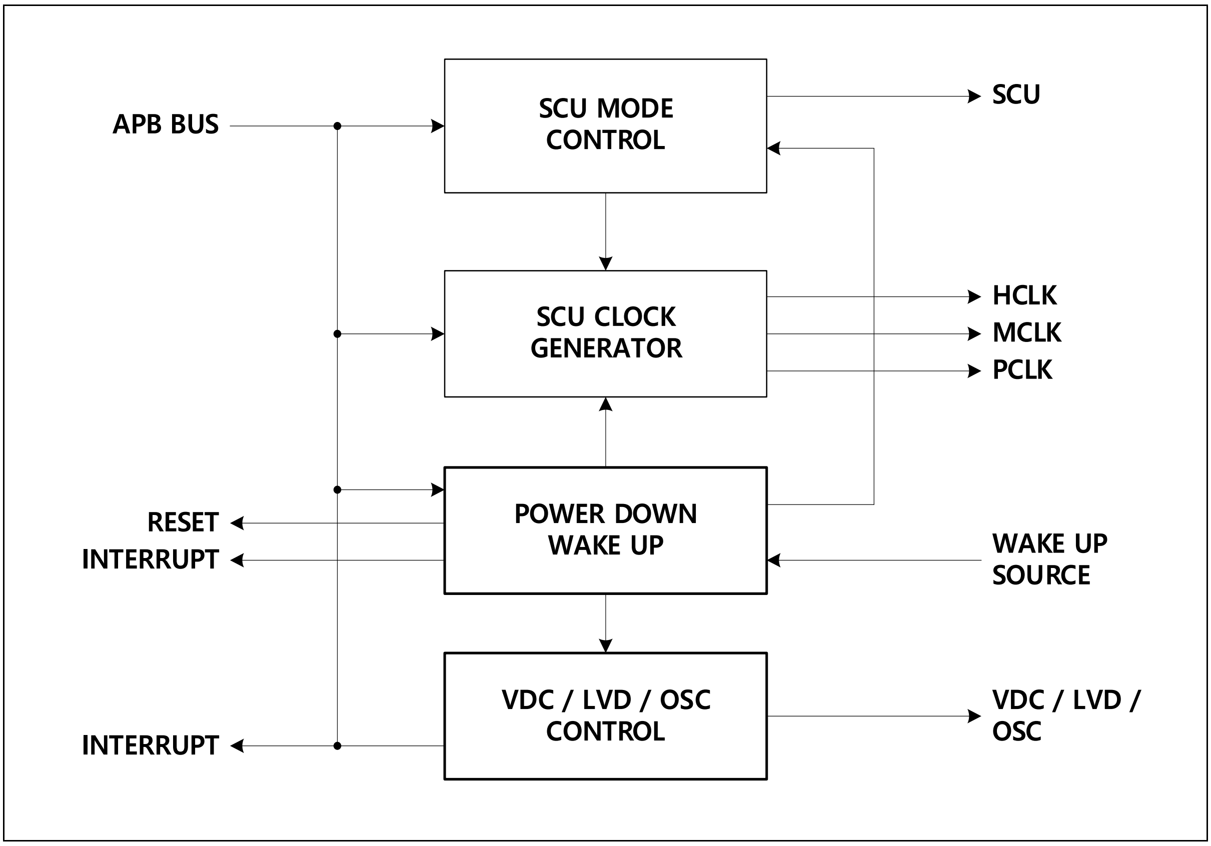
Table 6. SCU Pins

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

4.1 SCU block diagram

In this subsection, SCU block diagram is introduced in Figure 10.

Figure 10. SCU Block Diagram

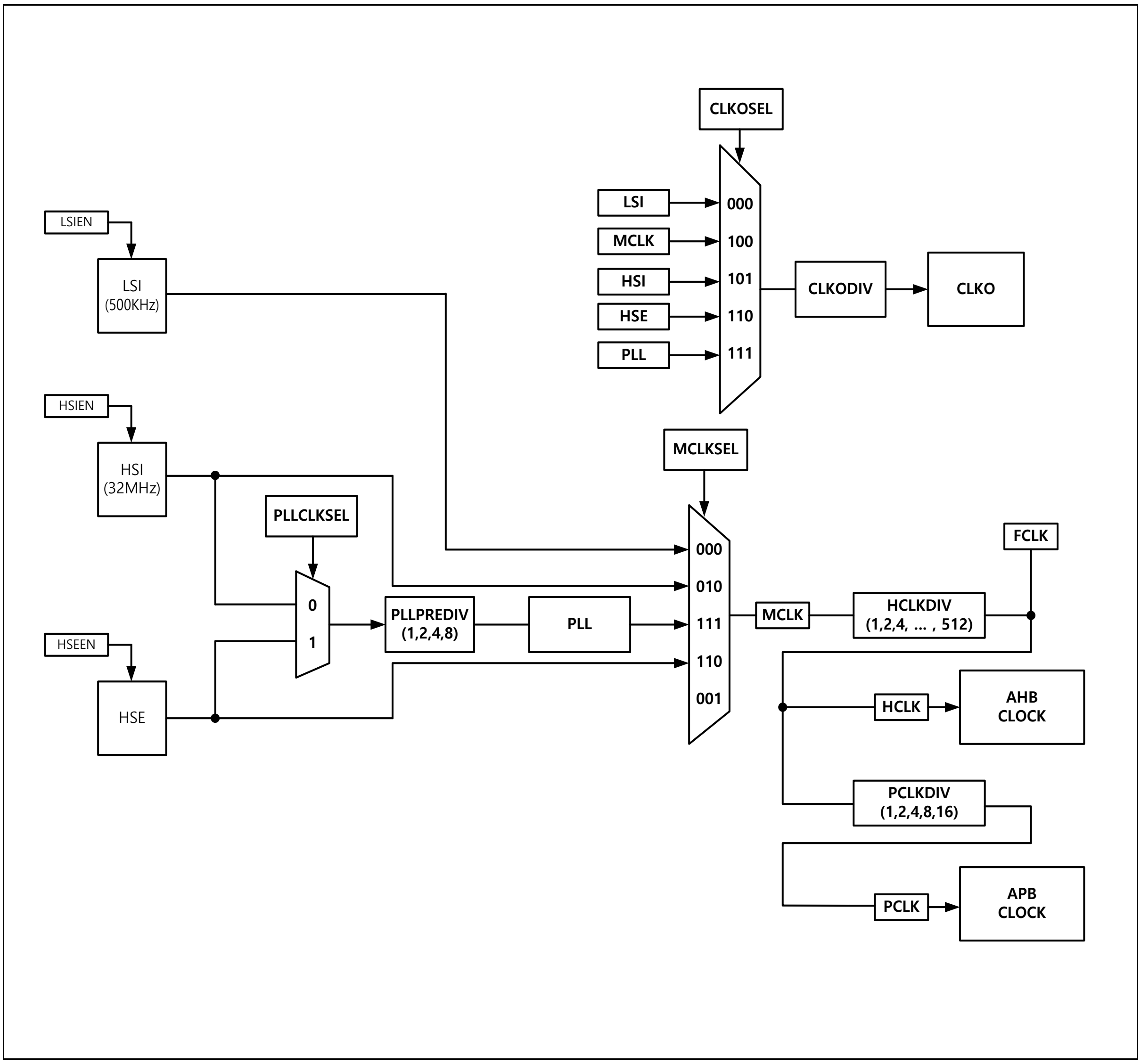


4.2 Clock system

A33M11x series has 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 11 and Table 7, users learn about the clock system of A33M11x devices and clock sources.

Figure 11. 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.

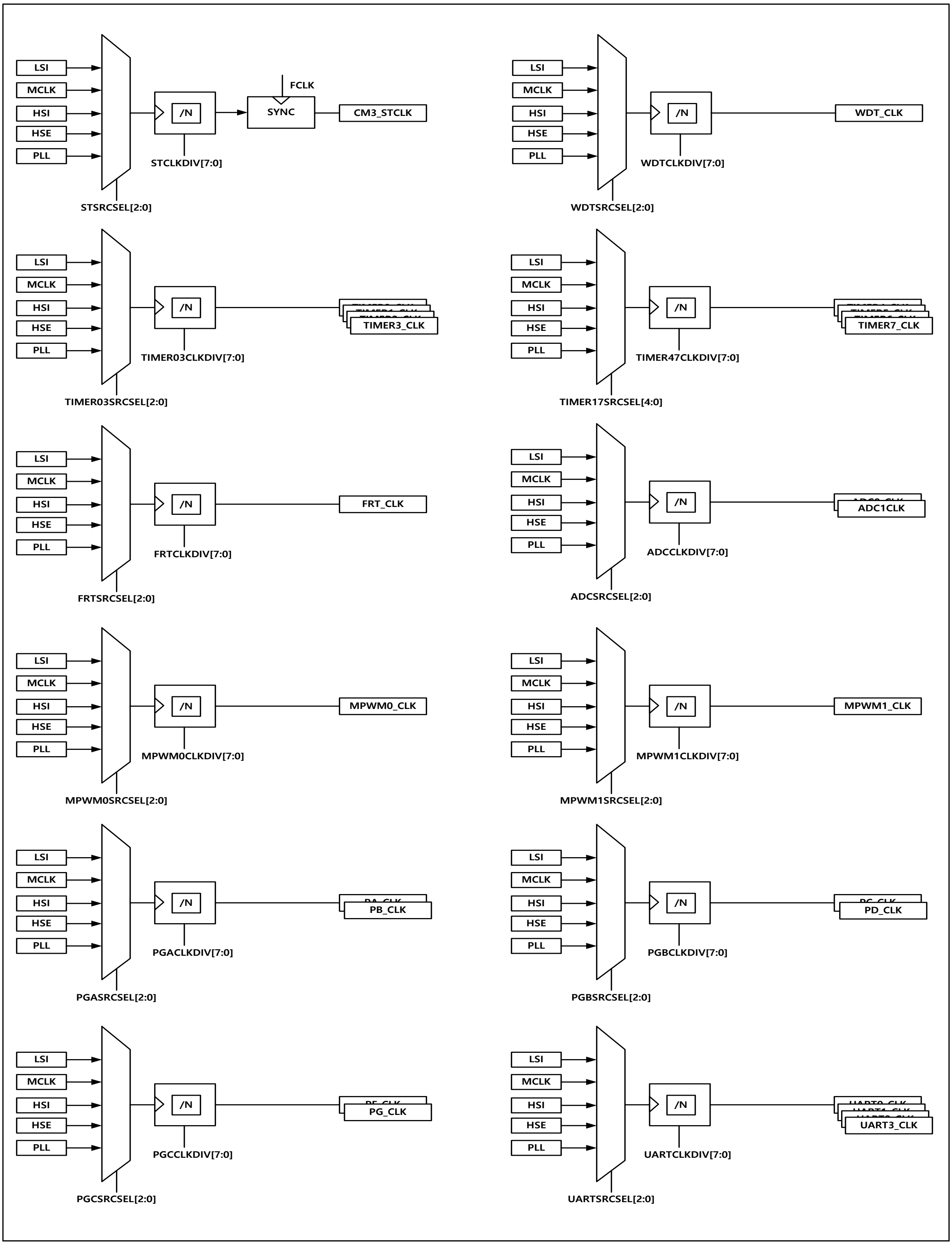
Table 7. Clock Sources

Clock name	Frequency	Description
HSE	4-16MHz	High Speed External Oscillator
PLL Clock	8-96MHz	On-chip PLL
HSI	32MHz	High Speed Internal OSC
LSI	500KHz	Low Speed Internal OSC

4.2.1 Configuration of miscellaneous clocks

The A33M11x series supports the “miscellaneous clocks” feature, which allows for assigning each peripheral with a different clock source (MCLK, HSE, PLL, LSI or HSI) at a different frequency division ratio. You can set each peripheral’s clock source and its frequency divider in the corresponding SCU_MCCR register. The supported division ratio can be one ranging from 1 to 255.

Figure 12. Configuration of Miscellaneous Clocks



4.2.2 HCLK clock domain

The HCLK is fed to the CPU and AHB. The Cortex-M3 CPU requires two clocks, the HCLK and FCLK. The FCLK stays enabled except in stop mode, whereas the HCLK can be disabled in idle mode.

The buses and memories are clocked by the HCLK. As the bus clock frequency is limited to a maximum of 96MHz, the HCLK frequency must not exceed 96MHz.

4.2.3 PCLK clock domain

The PCLK is used as a clock for any peripherals. Whether to enable or disable the PCLK for each peripheral is determined with the SCU_PCER registers; each peripheral block's registers cannot be read unless its PCLK input is enabled. And the PCLK stops operating in stop mode.

4.2.4 Clock configuration procedure

After the MCU is powered on, the LSI (500KHz) is initially enabled as the system clock source by default in the system operation sequence. Other clock sources are initially set by the user while the system is clocked by the LSI. The HSI (32MHz) can be enabled with SCU_CSCR (clock source control register). Before enabling the HSE block, the pin mux configuration should be set for XIN and XOUT. You must be careful not to affect other bits of PCC_MR and PCC_CR during this process. Once the HSE block has been enabled, you must wait for the crystal oscillation to stabilize.

The MCLK can be changed with SCU_SCCR (system clock source register). Figure 13 shows an example of how the system clock is changed.

※ If you change the main clock from LSI (500kHz) to PLL 80MHz or higher, it is recommended to change it gradually in the order shown below:

— Change order: LSI → HSE → PLL

① Once the MCU is powered on, the system clock source is LSI (500kHz).

② Initialize Data and instruction cache, and then set flash wait to 7.

③ Set XIN and XOUT on pin MUX to use the HSE.

④ Enable the HSE.

⑤ Set additional time of about 1ms for clock stabilization.

⑥ Enable the PLL.

⑦ Set additional time of about 1us for clock stabilization.

⑧ Set the PLL options. (Refer to **4.5.22** and **4.6.1**.)

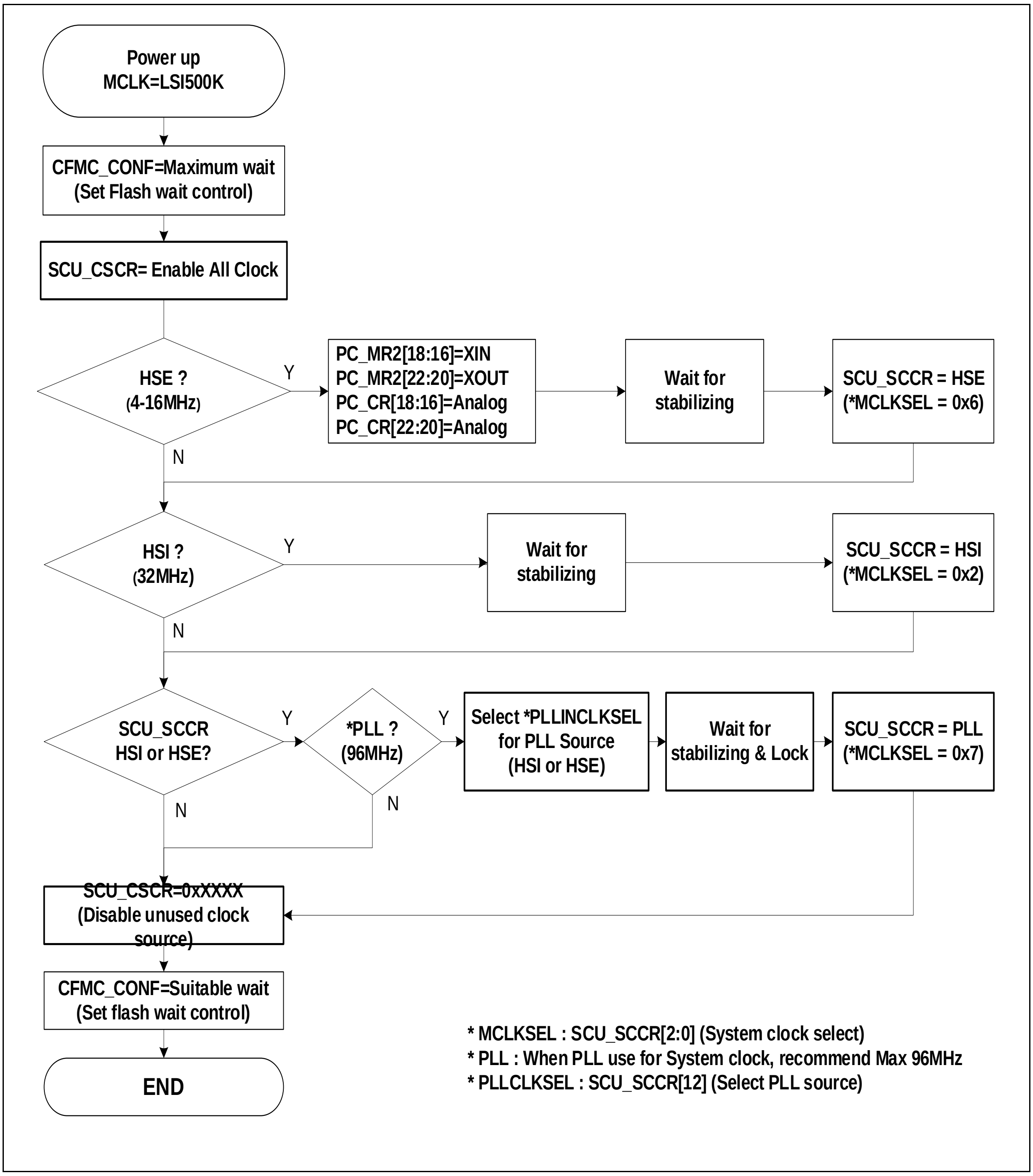
⑨ Set the HCLK to the MCLK divided by 2, and the PCLK to the HCLK.

⑩ Change the system clock source as the PLL.

⑪ Set the HCLK to the MCLK divided by 1, and the PCLK to the HCLK.

⑫ Set flash wait to 3.

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

FMCONF.WAIT	FLASH access wait	Available max. system clock frequency
0000	0-clock wait	Up to 28MHz
0001	1-clock wait	Up to 56MHz
0010	2-clock wait	Up to 84MHz
0011	3-clock wait	Up to 96MHz
0100	4-clock wait	Up to 96MHz
0101	5-clock wait	Up to 96MHz
0110	6-clock wait	Up to 96MHz
0111	7-clock wait	Up to 96MHz
1000	8-clock wait	Up to 96MHz
1001	9-clock wait	Up to 96MHz
1010	10-clock wait	Up to 96MHz
1011	11-clock wait	Up to 96MHz
1100	12-clock wait	Up to 96MHz
1101	13-clock wait	Up to 96MHz
1110	14-clock wait	Up to 96MHz
1111	15-clock wait	Up to 96MHz

Figure 14 shows how to set the peripheral clock. Selecting a peripheral clock is a typical method of MCCRn and PCLK.

Figure 14. Peripheral Clock Select: n = 1, 2, 3, 4, 5, 6 and 7

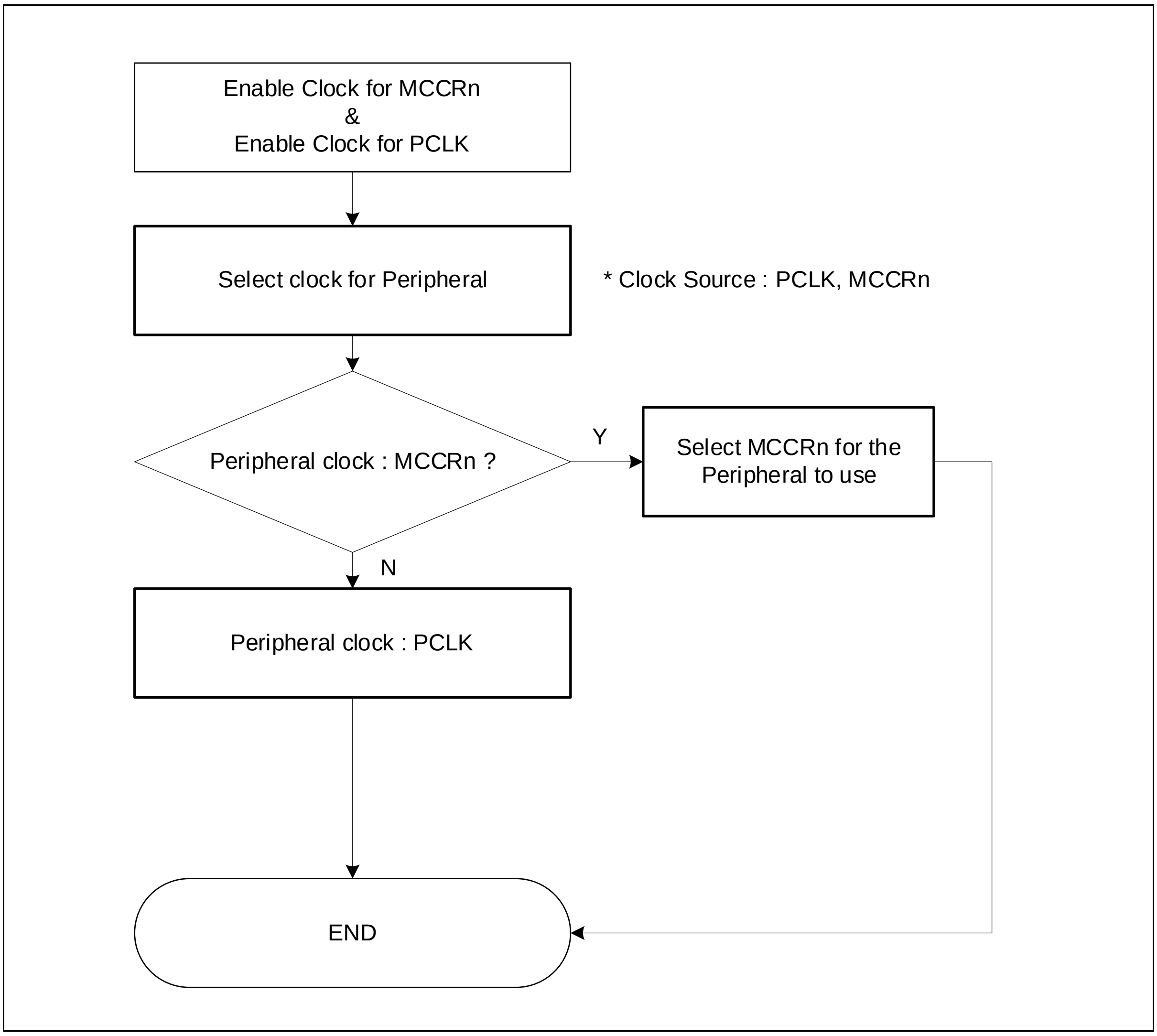

Enable Clock for MCCRn
&
Enable Clock for PCLK
Select clock for Peripheral
* Clock Source : PCLK, MCCRn
Peripheral clock : MCCRn ?
Y
Select MCCRn for the Peripheral to use
N
Peripheral clock : PCLK
END

Table 9. Peripheral Clock Select

Peripheral	MCCRn	PCLK
SYSTICK	MCCR1	O
WDT		O
MPWM0	MCCR2	N/A
MPWM1		N/A
TIMER03	MCCR3	O
TIMER47		O
ADC	MCCR4	O
PGAD		N/A
PGBD	MCCR5	N/A
PGCD		N/A
FRT	MCCR6	N/A
UART	MCCR7	N/A

4.3 Reset

The A33M11x series has two system reset options. One is a cold reset that is effective during power up or down sequence. The other is a 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	• POR	• nRESET Pin reset • LVR reset • WDT reset • MCLK Fail reset • HSE Fail reset • S/W reset • CPU request reset

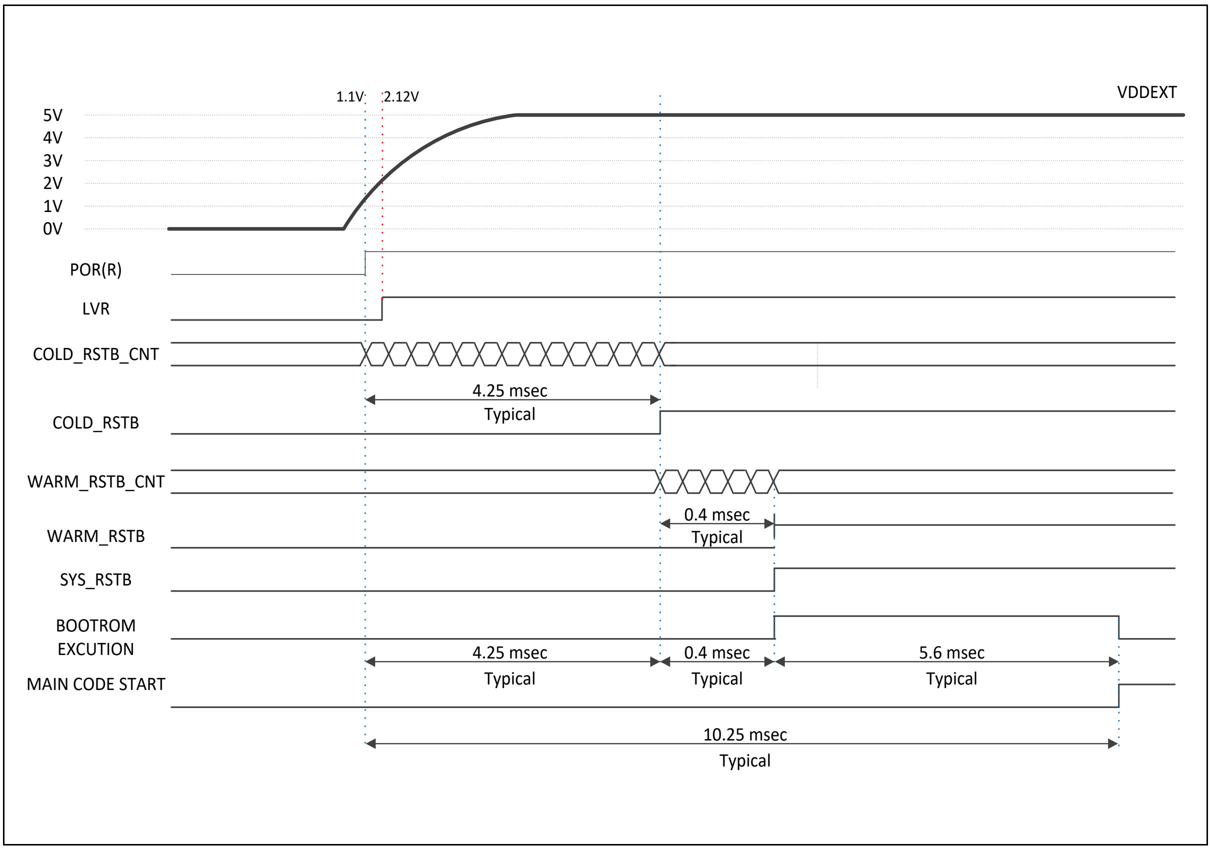
4.3.1 Cold reset

A cold reset plays an important role during a power-up process and affects the entire process of system booting. The internal VDC becomes active as soon as the External VDD is turn on. The internal POR is triggered when the External VDD is determined to reach 1.1V based on the amount of the internal VDC output. During the cold reset process, when the applied voltage exceeds 1.1V, the LSI clock is enabled.

After the stabilization of the internal VDC level for 4.25msec, the internal logic is initialized. And then, when the external VDD voltage rises above the LVR voltage level (2.12V), the cold reset is released and, after a waiting time of 0.4ms for warm reset synchronization, booting begins.

Figure 15 shows the power-up process and the initial reset waveforms.

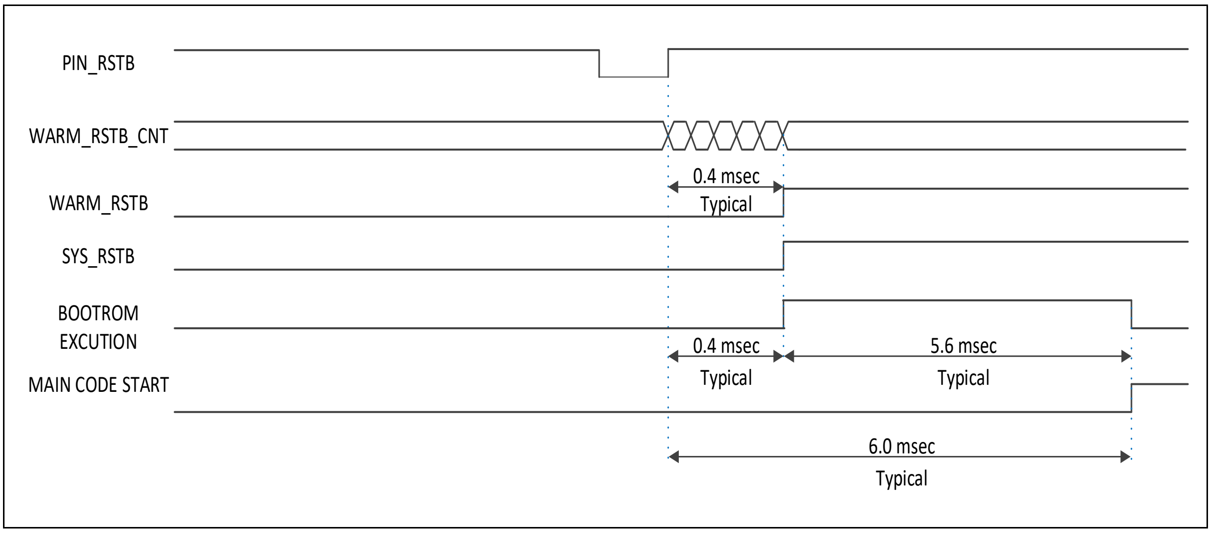
Figure 15. Power-up Procedure



4.3.2 Warm reset

A warm reset event is triggered for system initialization when the conditions of an internally set reset source are met. Warm reset sources are enabled or disabled by configuring SCU_RSER (reset source enable register), and their occurrence is written to SCU_RSSR (reset source status register). Which devices are to be initialized by a warm reset is determined by the settings of SCU_PRER (peripheral reset enable register). Using this register, a user can allow or disallow the initialization of each individual device.

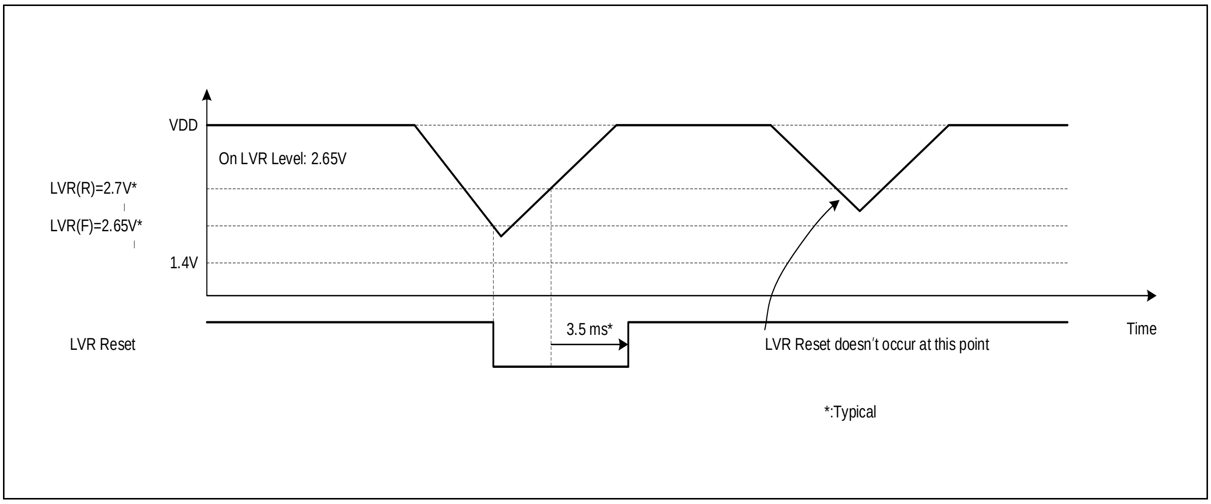
Figure 16. Warm Reset Diagram



4.3.3 LVR reset

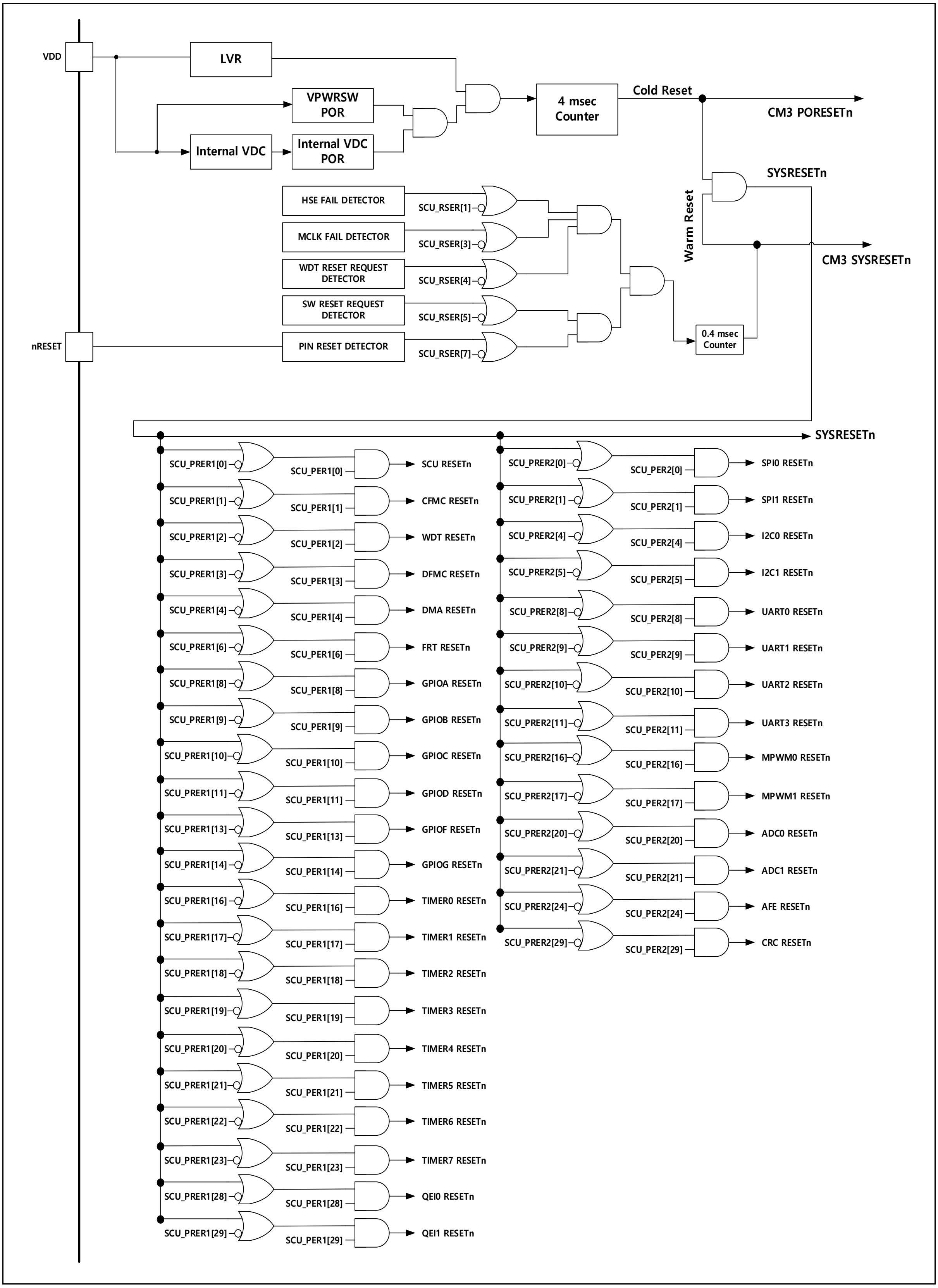
An LVR event is triggered when the operating voltage drops below a certain level during the MCU's operation. A user can choose to set the MCU to perform a reset or an interrupt when an LVR event is triggered. This low voltage reset is a warm reset. See the description on warm resetting for details.

Figure 17. LVR Reset Timing Diagram



4.3.4 Reset tree

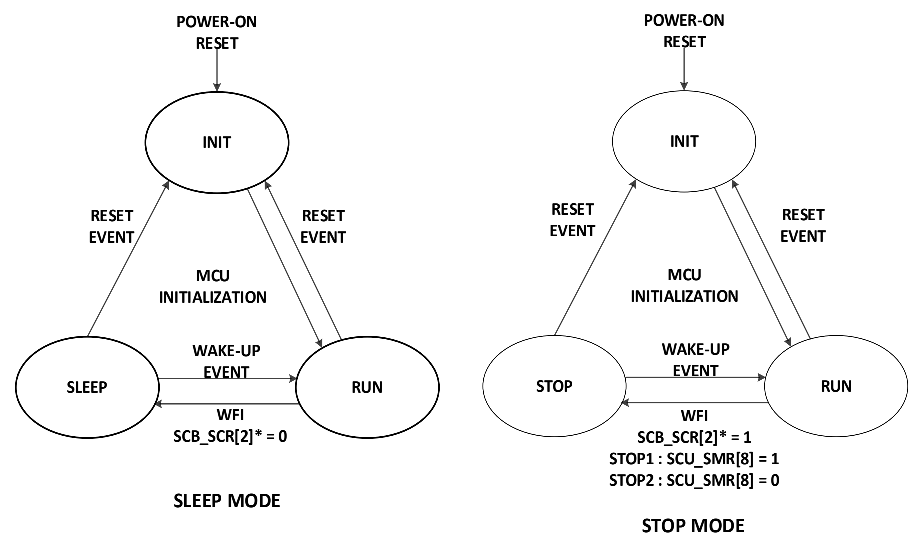
Figure 18. Reset Tree Configuration



4.4 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 two power down modes (STOP1, STOP2) 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 19 describes transition between the operation modes.

Figure 19. Transition between Operation Modes



NOTE: * means that SCB_SCR is System Register in System Control Block.

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
STOP1	WFI (Wait for Interrupt): SCB_SCR[2]*=1, SCU_SMR[8] =1	RUN	INIT
STOP2	WFI (Wait for Interrupt): SCB_SCR[2]*=1, SCU_SMR[8] =0	RUN	INIT

NOTE: SCB_SCR is System Control Register in System Control Block.

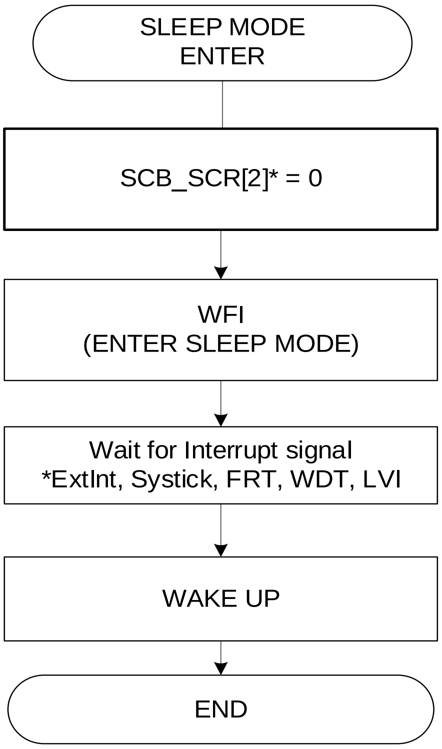
4.4.1 RUN mode

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 RUN mode.

4.4.2 SLEEP mode

Once the MCU enters in SLEEP mode, the CPU becomes inactive. By setting the PER and the PCER registers, a user can determine which peripherals are to be inactive in SLEEP mode.

Figure 20. SLEEP Mode Operation Sequence



NOTE: SCB_SCR is System Control Register in System Control Block.

4.4.3 STOP mode

When the core goes under stop state using WFI instruction, the chip enters in STOP mode. STOP mode is divided into STOP1 and STOP2 modes.

To enter STOP1 mode, first set SCU_SMR[8]=1 and use the WFI command. In STOP1 mode, all peripherals except WDT and FRT of internal VDC domain are in power off state. To enter STOP2 mode, first set SCU_SMR[8]=0 and use the WFI command. In STOP2 mode, all peripherals of internal VDC domain are stopped. To wake up in STOP mode, an interrupt request must be generated and can be selected in the SCU_WUER register. Stabilization time is 4ms after wakeup event occurs.

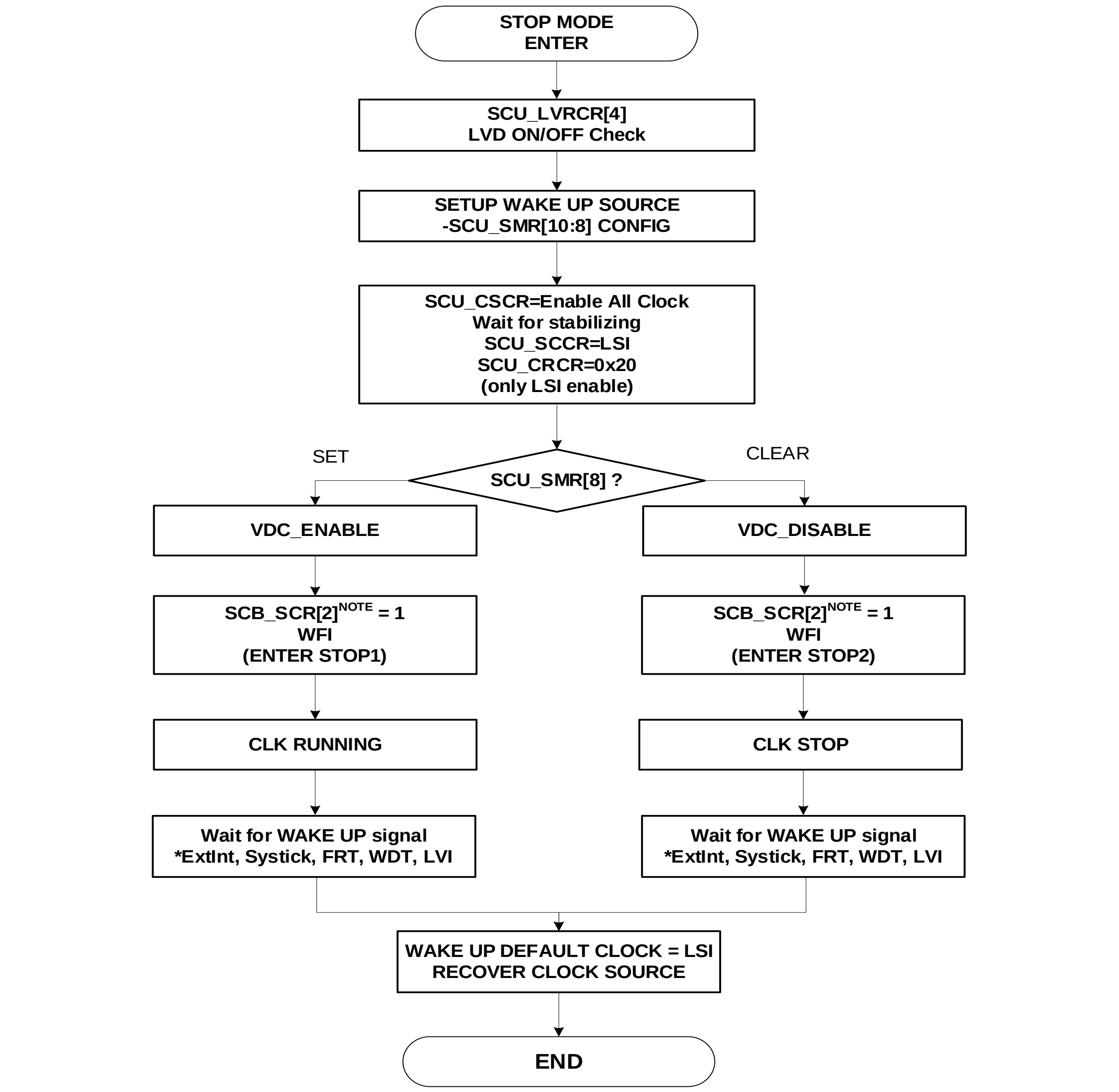
When entering STOP mode, LSI500K is automatically selected as the MCLK, HSE is selected as the PLL input clock, and all clock sources are automatically disabled. These are maintained after wakeup. If the SCU_SMR register is used, a function that automatically disables system clocks such as HSE, PLL, HSI, and LSI may not be used.

Table 12. STOP Mode Configuration

Mode	Condition	Wakeup source
STOP1	WFI SCB_SCR[2]* = 1 SCU_SMR[8] = 1	Source included in WUER
STOP2	WFI SCB_SCR[2]* = 1 SCU_SMR[8] = 0	Source included in WUER

NOTE: SCB_SCR is System Control Register in System Control Block.

Figure 21. STOP Mode Operation Sequence



NOTE: SCB_SCR is System Control Register in System Control Block.

Table 13 lists configurable clocks in each operating mode.

Table 13. Configurable Clocks in Each Operating Mode

Mode	Core	Peri.	IP				
			VDC	PLL	LSI	HSI	HSE
RUN	ON	User-defined	ON	User-defined	User-defined	User-defined	User-defined
SLEEP	OFF	User-defined	ON	User-defined	User-defined	User-defined	User-defined
STOP1	OFF	Only WDT, FRT on	ON	User-defined	User-defined	User-defined	User-defined
STOP2	OFF	All off	OFF	OFF	User Define	OFF	OFF

4.5 Registers

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

Table 14. Base Address of SCU: Chip Configuration

Name	Base address
SCU (Chip Configuration)	0x4000_F000

Table 15. SCU Register Map: Chip Configuration

Name	Offset	Type	Description	Reset value	Reference
CHIPCONFIG_VENDORID	0x0000	RO^{NOTE}	Vendor Identification Register	0x4142_4F56	4.5.1
CHIPCONFIG_CHIPID	0x0004	RO^{NOTE}	Chip Identification Register.	0x4D31_A00x	4.5.2
CHIPCONFIG_REVNR	0x0008	RO^{NOTE}	Revision Number Register	0x0000_00xx	4.5.3

NOTE: ‘RO’ means ‘Read Only’.

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

Table 16. Base Address of SCU

Name	Base address
SCU	0x4000_0000

Table 17. SCU Register Map

Name	Offset	Type	Description	Reset value	Reference
SCU_SMR	0x0004	RW	System mode register	0x0000_0030	4.5.4
SCU_SRCR	0x0008	RW	System reset control register	0x0000_0000	4.5.5
SCU_WUER	0x0010	RW	Wake-up source enable register	0x0000_0000	4.5.6
SCU_WUSR	0x0014	RO	Wake-up source status register	0x0000_0000	4.5.7
SCU_RSER	0x0018	RW	Reset source enable register	0x0000_00D1	4.5.8
SCU_RSSR	0x001C	RC	Reset source status register	0x0000_0101*	4.5.9
SCU_PRER1	0x0020	RW	Peripheral reset enable register 1	0x30FF_6F5F*	4.5.10
SCU_PRER2	0x0024	RW	Peripheral reset enable register 2	0x2133_0F33*	4.5.11
SCU_PER1	0x0028	RW	Peripheral enable register 1	0x0000_6F0F*	4.5.12
SCU_PER2	0x002C	RW	Peripheral enable register 2	0x0000_0100*	4.5.13
SCU_PCER1	0x0030	RW	Peripheral clock enable register 1	0x0000_6F0F*	4.5.14
SCU_PCER2	0x0034	RW	Peripheral clock enable register 2	0x0000_0100*	4.5.15

Table 17. SCU Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
SCU_CSCR	0x0040	RW	Clock source control register	0x0000_0020	4.5.16
SCU_SCCR	0x0044	RW	System clock control register	0x0000_0000	4.5.17
SCU_CMR	0x0048	RW	Clock monitor register	0x0000_0000	4.5.18
SCU_COR	0x0050	RW	Clock output register	0x0000_000F	4.5.19
SCU_NMICR	0x0054	RW	Non-maskable interrupt control register	0x0000_0000	4.5.20
SCU_NMISR	0x0058	RC	Non-maskable interrupt status register	0x0000_0000	4.5.21
SCU_PLLCON	0x0060	RW	PLL control register	0x0600_0000	4.5.22
SCU_VDCCON	0x0064	RW	VDC control register	0x0000_007F	4.5.23
SCU_LVICR	0x0068	RW	LVI control register	0x0000_0000	4.5.24
SCU_LVISR	0x006C	RC	LVI status register	0x0000_0000	4.5.25
SCU_LVRCR	0x0070	RW	LVR control register	0x0000_0005	4.5.26
SCU_EOSCR	0x0080	RW	External oscillator control register	0x0000_0000	4.5.27
SCU_MCCR1	0x0090	RW	MISC clock control register 1	0x0001_0000	4.5.28
SCU_MCCR2	0x0094	RW	MISC clock control register 2	0x0000_0000	4.5.29
SCU_MCCR3	0x0098	RW	MISC clock control register 3	0x0000_0000	4.5.30
SCU_MCCR4	0x009C	RW	MISC clock control register 4	0x0000_0000	4.5.31
SCU_MCCR5	0x00A0	RW	MISC clock control register 5	0x0000_0000	4.5.32
SCU_MCCR6	0x00A4	RW	MISC clock control register 6	0x0000_0000	4.5.33
SCU_MCCR7	0x00A8	RW	MISC clock control register 7	0x0001_0000	4.5.34
SCU_SYSTEN	0x00F0	WO	System access enable register	0x0000_0000	4.5.35

NOTES:

1. ‘RO’ means ‘Read Only’.
2. ‘RW’ means ‘Read and Write’.
3. ‘WO’ means ‘Write Only’.
4. ‘RC’ means ‘Read and Write 1 Clear’.

4.5.1 CHIPCONFIG_VENDORID: vendor ID register

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

CHIPCONFIG_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

4.5.2 CHIPCONFIG_CHIPID: chip ID register

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

CHIPCONFIG_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			
0xXXXX_XXXX			
RO			

31 0	CHIPID	Chip Identification bits.	
		0x4D33_A004	256Kbyte Flash memory/16Kbyte RAM
		0x4D33_A005	128Kbyte Flash memory/16Kbyte RAM

4.5.3 CHIPCONFIG_REVNR: revision number register

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

CHIPCONFIG_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.

4.5.4 SCU_SMR: system mode register

The SCU_SMR register contains bits controllable in stop mode and informs of which operating mode the MCU had been in before it returned to initial mode. Once a reset event occurs, the previous operating mode is recorded in this register. The register is 32 bits wide.

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	HSEAON	PLLAON	HSIAON	LSIAON	VDCAON	Reserved	PREVMODE	Reserved
-	0	0	0	0	0	-	11	-
-	RW	RW	RW	RW	RW	-	RO	-

12	HSEAON	HSE enablement in stop mode	
		0	Automatically disables the HSE in stop mode.
		1	Leaves the HSE enabled in stop mode.
11	PLLAON	PLL enablement in stop mode	
		0	Automatically disables the PLL in stop mode.
		1	Leaves the PLL enabled in stop mode.
10	HSIAON	HSI enablement in stop mode	
		0	Automatically disables the HSI in stop mode.
		1	Leaves the HSI enabled in stop mode.
9	LSIAON	LSI enablement in stop mode	
		0	Automatically disables the LSI in stop mode.
		1	Leaves the LSI enabled in stop mode.
8	VDCAON	VDC enablement in stop mode	
		0	Automatically disables the VDC in stop mode.
		1	Leaves the VDC enabled in stop mode.
5 4	PREVMODE	The operating mode when the last reset event occurred	
		00	The MCU was in run mode.
		01	The MCU was in sleep mode.
		10	The MCU was in stop mode.
		11	The MCU was in initial mode.

NOTES:

1. Even if you set the SMR register for a clock to stay enabled during stop mode, the clock will not run if it has been set disabled in the SCU_CSCR register.
2. If the PLLAON bit is set to 1, the VDC stays active regardless of the setting of the VDCAON bit.

4.5.5 SCU_SRCR: system reset control register

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

SCU_SRCR=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
Reserved																															SWRST
-																															0
-																															WO

0	SWRST	Internal soft reset activation bit (check RSER[5] for reset)	
		0	Normal operation
		1	Internal soft resets can occur (The bit becomes automatically cleared).

4.5.6 SCU_WUER: wakeup source enable register

The SCU_WUER register is used when the MCU is in stop mode or sleep mode. To enable a signal to be used as a wake-up source, you must set the corresponding bit to 1. And the bits for those not used as a wake-up source must be set to 0. This register is 32 bits wide.

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	SYSTICKWUE	Reserved	GPIOGWUE	GPIOFWUE	Reserved	GPIODWUE	GPIOCWUE	GPIOBWUE	GPIOAWUE	Reserved	FRTWUE	WDTWUE	LVIWUE
-	0	-	0	0	-	0	0	0	0	-	0	0	0
-	RW	-	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW

16	SYSTICKWUE	Whether or not to use the Systick Timer event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
14	GPIOGWUE	Whether or not to use the GPIOG port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
13	GPIOFWUE	Whether or not to use the GPIOF port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
11	GPIODWUE	Whether or not to use the GPIOD port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
10	GPIOCWUE	Whether or not to use the GPIOC port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
9	GPIOBWUE	Whether or not to use the GPIOB port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
8	GPIOAWUE	Whether or not to use the GPIOA port event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
2	FRTWUE	Whether or not to use the FRT event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
1	WDTWUE	Whether or not to use the WDT event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.
0	LVIWUE	Whether or not to use the LVI event as a wake-up source	
		0	Does not use the event as a wake-up source.
		1	Uses the event as a wake-up source.

NOTE: When using a wake-up source in stop mode, the clock source activation of the peripheral device must be set in the SCU_SMR register.

4.5.7 SCU_WUSR: wakeup source status register

When the MCU is woken up by a wake-up source, the corresponding bit in the SCU_WUSR is flagged. A bit set to 1 means that its corresponding wake-up event has occurred. It is a read-only register. When an event source is cleared, the corresponding bit in this register is also cleared.

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	SYSTICKWU	Reserved	GPIOGWU	GPIOFWU	Reserved	GPIODWU	GPIOCWU	GPIOBWU	GPIOAWU	Reserved	FRTWU	WDTWU	LVIWU
-	0	-	0	0	-	0	0	0	0	-	0	0	0
-	RW	-	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW

16	SYSTICKWU	Whether or not the Systick timer event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
14	GPIOGWU	Whether or not the GPIOG port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
13	GPIOFWU	Whether or not the GPIOF port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
11	GPIODWU	Whether or not the GPIOD port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
10	GPIOCWU	Whether or not the GPIOC port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
9	GPIOBWU	Whether or not the GPIOB port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
8	GPIOAWU	Whether or not the GPIOA port event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
2	FRTWU	Whether or not the FRT event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
1	WDTWU	Whether or not the WDT event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.
0	LVIWU	Whether or not the LVI event has been triggered for wake-up	
		0	The wake-up event has not been triggered.
		1	The wake-up event has been triggered.

4.5.8 SCU_RSER: reset source enable register

The SCU_RSER register allows you to configure input signals that trigger a reset event. Setting a bit to 1 enables its corresponding reset signal to trigger a reset; setting it to 0 disables the reset signal, thereby masking 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			LOCKUPRST	Reserved	PINRST	CPURST	SWRST	WDTRST	MCLKRST	Reserved	HSERST	LVDRST
-			0	-	1	1	0	1	0	-	0	1
-			RW	-	RW	RW	RW	RW	RW	-	RW	RW

9	LOCKUPRST	Whether to enable or disable the CPU lock-up reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
7	PINRST	Whether to enable or disable the external-pin reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
6	CPURST	Whether to enable or disable the CPU request reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
5	SWRST	Whether to enable or disable the software reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
4	WDTRST	Whether to enable or disable the WDT reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
3	MCKFRST	Whether to enable or disable the MCLK error reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
1	HSEFRST	Whether to enable or disable the HSE error reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.
0	LVDRST	Whether to enable or disable the LVR reset signal	
		0	Disables the signal to trigger a reset event.
		1	Enables the signal to trigger a reset event.

4.5.9 SCU_RSSR: reset source status register

The SCU_RSSR register records occurrences of reset events. A bit read as 1 means that its corresponding reset source has triggered a reset.

To clear the flag of a reset source, write a 1 to the flag bit. It is a 32-bit register.

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			LOCKUPRST	PORST	PINRST	CPURST	SWRST	WDTRST	MCLKRST	Reserved	HSERST	LVDRST
-			0	1	0	0	0	0	0	-	0	1
-			RC	RC	RC	RC	RC	RC	RC	-	RC	RC

9	LOCKUPRST	Whether or not the CPU lock-up reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
8	PORST	Whether or not the POR reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
7	PINRST	Whether or not the external pin reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
6	CPURST	Whether or not the CPU core reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
5	SWRST	Whether or not the software reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
4	WDTRST	Whether or not the WDT reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
3	MCLKFRST	Whether or not the MCLK error reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.
1	HSEFRST	Whether or not the HSE error reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred.

			Write: Clears the flag.
0	LVDRST	Whether or not the LVD reset has occurred	
		0	Read: The reset has not occurred. Write: N/A
		1	Read: The reset has occurred. Write: Clears the flag.

NOTE: When reset source is founded, write ‘1’ into the corresponding bit to clear the reset status.

4.5.10 SCU_PRER1: peripheral reset enable register 1

The SCU_PRER registers determine whether or not allow each peripheral to be initialized when a warm reset event occurs. They consist of PRER1 and PRER2. Once a reset event occurs, the peripherals whose corresponding bits are set to 1 are initialized and those set to 0 remain uninitialized. However, the POR reset initializes all peripherals.

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
Reserved		QEI1	QEI0	Reserved				TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	Reserved	GPIOG	GPIOF	Reserved	GPIOD	GPIOC	GPIOB	GPIOA	Reserved	FRT	Reserved	DMA	DFMC	WDT	CFMC	SCU
-		1	1	-				1	1	1	1	1	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	RW	-	RW	-	RW	RW	RW	RW	RW

29	QEI1	QEI1 reset mask
28	QEI0	QEI0 reset mask
23	TIMER7	TIMER7 reset mask
22	TIMER6	TIMER6 reset mask
21	TIMER5	TIMER5 reset mask
20	TIMER4	TIMER4 reset mask
19	TIMER3	TIMER3 reset mask
18	TIMER2	TIMER2 reset mask
17	TIMER1	TIMER1 reset mask
16	TIMER0	TIMER0 reset mask
14	GPIOG	GPIOG reset mask
13	GPIOF	GPIOF reset mask
11	GPIOD	GPIOD reset mask
10	GPIOC	GPIOC reset mask
9	GPIOB	GPIOB reset mask
8	GPIOA	GPIOA reset mask
6	FRT	FRT reset mask
4	DMA	DMA reset mask
3	DFMC	Data Flash memory controller (DFMC) reset mask
2	WDT	WDT reset mask
1	CFMC	Code Flash memory controller reset mask
0	SCU	SCU reset mask

4.5.11 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
Reserved	CRC	Reserved	AFE	Reserved	ADC1	ADC0	Reserved	MPWM1	MPWM0	Reserved	UART3	UART2	UART1	UART0	Reserved	I2C1	I2C0	Reserved	SPI1	SPI0
-	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

29	CRC	CRC reset mask
24	AFE	AFE reset mask
21	ADC1	ADC1 reset mask
20	ADC0	ADC0 reset mask
17	MPWM1	MPWM1 reset mask
16	MPWM0	MPWM0 reset mask
11	UART3	UART3 reset mask
10	UART2	UART2 reset mask
9	UART1	UART1 reset mask
8	UART0	UART0 reset mask
5	I2C1	I^{2}C1 reset mask
4	I2C0	I^{2}C0 reset mask
1	SPI1	SPI1 reset mask
0	SPI0	SPI0 reset mask

4.5.12 SCU_PER1: peripheral enable register1

To enable a peripheral, you must write a 1 to its corresponding bit in PER1 or 2. Before it is enabled, the peripheral is reset.

Because only a certain set of peripherals are enabled at the initial reset, an additional action is required to have other peripherals enabled if you want to use them. By writing a 0 to the bits representing the peripherals you'll not use, you can disable them.

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
Reserved		QEI1	QEI0	Reserved				TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	Reserved	GPIOG	GPIOF	Reserved	GPIOD	GPIOC	GPIOB	GPIOA	Reserved	FRT	Reserved	DMA	Reserved			
-		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	-			

29	QEI1	Whether to enable or disable QEI1
28	QEI0	Whether to enable or disable QEI0
23	TIMER7	Whether to enable or disable TIMER7
22	TIMER6	Whether to enable or disable TIMER6
21	TIMER5	Whether to enable or disable TIMER5
20	TIMER4	Whether to enable or disable TIMER4
19	TIMER3	Whether to enable or disable TIMER3
18	TIMER2	Whether to enable or disable TIMER2
17	TIMER1	Whether to enable or disable TIMER1
16	TIMER0	Whether to enable or disable TIMER0
14	GPIOG	Whether to enable or disable GPIOG
13	GPIOF	Whether to enable or disable GPIOF
11	GPIOD	Whether to enable or disable GPIOD
10	GPIOC	Whether to enable or disable GPIOC
9	GPIOB	Whether to enable or disable GPIOB
8	GPIOA	Whether to enable or disable GPIOA
6	FRT	Whether to enable or disable FRT
4	DMA	Whether to enable or disable DMA

4.5.13 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
Reserved	CRC	Reserved	AFE	Reserved	ADC1	ADC0	Reserved	MPWM1	MPWM0	Reserved	UART3	UART2	UART1	UART0	Reserved	I2C1	I2C0	Reserved	SPI1	SPI0
-	0	-	0	-	0	0	-	0	0	-	0	0	0	1	-	0	0	-	0	0
-	RW	-	RW	-	RW	RW	-	RW	RW	-	RW	RW	RW	RW	-	RW	RW	-	RW	RW

29	CRC	Whether to enable or disable CRC
24	AFE	Whether to enable or disable AFE
21	ADC1	Whether to enable or disable ADC1
20	ADC0	Whether to enable or disable ADC0
17	MPWM1	Whether to enable or disable MPWM1
16	MPWM0	Whether to enable or disable MPWM0
11	UART3	Whether to enable or disable UART3
10	UART2	Whether to enable or disable UART2
9	UART1	Whether to enable or disable UART1
8	UART0	Whether to enable or disable UART0
5	I2C1	Whether to enable or disable I^{2}C1
4	I2C0	Whether to enable or disable I^{2}C0
1	SPI1	Whether to enable or disable SPI1
0	SPI0	Whether to enable or disable SPI0

4.5.14 SCU_PCER1: peripheral clock enable register 1

To enable the clock to a peripheral, you must write a 1 to its corresponding bit in PCER1 or 2. A peripheral cannot reset or operate normally until it is clocked.

If a peripheral will not be used, you must write a 0 to its corresponding PCER1/2 bit to disable its clock. When the MCU is in sleep or stop mode, the clock is fed only to active peripherals.

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
Reserved	QEI1	QEI0	Reserved	TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	Reserved	GPIOG	GPIOF	Reserved	GPIOD	GPIOC	GPIOB	GPIOA	Reserved	FRT	Reserved	DMA	Reserved
-	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	-

29	QEI1	Whether to enable or disable the clock to QEI1
28	QEI0	Whether to enable or disable the clock to QEI0
23	TIMER7	Whether to enable or disable the clock to TIMER7
22	TIMER6	Whether to enable or disable the clock to TIMER6
21	TIMER5	Whether to enable or disable the clock to TIMER5
20	TIMER4	Whether to enable or disable the clock to TIMER4
19	TIMER3	Whether to enable or disable the clock to TIMER3
18	TIMER2	Whether to enable or disable the clock to TIMER2
17	TIMER1	Whether to enable or disable the clock to TIMER1
16	TIMER0	Whether to enable or disable the clock to TIMER0
14	GPIOG	Whether to enable or disable the clock to GPIOG
13	GPIOF	Whether to enable or disable the clock to GPIOF
11	GPIOD	Whether to enable or disable the clock to GPIOD
10	GPIOC	Whether to enable or disable the clock to GPIOC
9	GPIOB	Whether to enable or disable the clock to GPIOB
8	GPIOA	Whether to enable or disable the clock to GPIOA
6	FRT	Whether to enable or disable the clock to FRT
4	DMA	Whether to enable or disable the clock to DMA

4.5.15 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
Reserved	CRC	Reserved	AFE	Reserved	ADC1	ADC0	Reserved	MPWM1	MPWM0	Reserved	UART3	UART2	UART1	UART0	Reserved	I2C1	I2C0	Reserved	SPI1	SPI0
-	0	-	0	-	0	0	-	0	0	-	0	0	0	1	-	0	0	-	0	0
-	RW	-	RW	-	RW	RW	-	RW	RW	-	RW	RW	RW	RW	-	RW	RW	-	RW	RW

29	CRC	Whether to enable or disable the clock to CRC
24	AFE	Whether to enable or disable the clock to AFE
21	ADC1	Whether to enable or disable the clock to ADC1
20	ADC0	Whether to enable or disable the clock to ADC0
17	MPWM1	Whether to enable or disable the clock to MPWM1
16	MPWM0	Whether to enable or disable the clock to MPWM0
11	UART3	Whether to enable or disable the clock to UART3
10	UART2	Whether to enable or disable the clock to UART2
9	UART1	Whether to enable or disable the clock to UART1
8	UART0	Whether to enable or disable the clock to UART0
5	I2C1	Whether to enable or disable the clock to I^{2}C1
4	I2C0	Whether to enable or disable the clock to I^{2}C0
1	SPI1	Whether to enable or disable the clock to SPI1
0	SPI0	Whether to enable or disable the clock to SPI0

4.5.16 SCU_CSCR: clock source control register

The A33M11x series is equipped with a number of different clock sources for generating internal clock signals. By setting the SCU_CSCR register, each of the clock sources can be enabled. It is a 32-bit 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
Reserved					LSICON	Reserved	HSICON	Reserved	HSECON	Reserved
-					1	-	0	-	0	-
-					RW	-	RW	-	RW	-

5	LSICON	Low speed internal oscillator control	
		0	Disables the low speed internal oscillator.
		1	Enables the low speed internal oscillator.
3	HSICON	High speed internal oscillator control	
		0	Disables the high speed internal oscillator.
		1	Enables the high speed internal oscillator.
1	HSECON	External crystal main oscillator control	
		0	Disables the external main crystal oscillator.
		1	Enables the external main crystal oscillator.

4.5.17 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
Reserved	HCLKDIV	Reserved	PCLKDIV	Reserved	PLLCLKSEL	Reserved	PLLPREDIV	Reserved	MCLKSEL
-	0000	-	000	-	0	-	00	-	000
-	RW	-	RW	-	RW	-	RW	-	RW

27 24	HCLKDIV	Clock divider for HCLK input	
		0000	HCLK = MCLK
		0001	HCLK = MCLK / 2
		0010	HCLK = MCLK / 4
		0011	HCLK = MCLK / 8
		0100	HCLK = MCLK / 16
		0101	HCLK = MCLK / 32
		0110	HCLK = MCLK / 64
		0111	HCLK = MCLK / 128
		1000	HCLK = MCLK / 256
		1001	HCLK = MCLK / 512
		Other	Reserved
18 16	PCLKDIV	Clock divider for PCLK input	
		000	PCLK = HCLK
		001	PCLK = HCLK / 2
		010	PCLK = HCLK / 4
		011	PCLK = HCLK / 8
		100	PCLK = HCLK / 16
		Other	Reserved
12	PLLCLKSEL	PLL clock selection	
		0	Selects the HSI as the clock source for the PLL.
		1	Selects the HSE as the clock source for the PLL.
9 8	PLLPREDIV	Clock divider for PLL input	
		00	PLLINCLK = PLLCLKSEL / 1
		01	PLLINCLK = PLLCLKSEL / 2
		10	PLLINCLK = PLLCLKSEL / 4
		11	PLLINCLK = PLLCLKSEL / 8
2 0	MCLKSEL	System clock selection	
		000	LSI
		010	HSI
		110	HSE
		111	PLL
		Others	Reserved

NOTES:

1. When change MCLKSEL, both clock sources should be alive. Ex) Both of HSI and HSE should be alive, otherwise the chip will malfunction.
2. If MCLKSEL is set to HSE, PC12 and PC13 must be set to XIN and XOUT functions.

3. When change PLLCLKSEL, both the HSI and HSE must be enable.
4. For the analog outputs, the PxMux bits of the Pn_MR1 and Pn_MR2 registers must be set to b’111 (Function 4) and the Px bit of the Pn_CR register must be set to b’1x (input).
   If the Px bit of the Pn_CR register is set to b’00 (push-pull output) or b’01 (open-drain output), the port may not operate correctly.
   Analog output pins are listed below:
   PA0(AO0), PA7(AO1), PA8(AO2), PA15(AO3), PC13(XOUT)

4.5.18 SCU_CMR: clock monitoring register

The LSI can be used to monitor internal clocks for protection purposes. The SCU_CMR register is 32 bits wide.

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	MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	HSEMNT	HSEIE	HSEFAIL	HSESTS
-		0	-	0	0	0	0	0	0	0	0
0		RW	0	RW	RW	RC	RO	RW	RW	RC	RO

15	MCLKREC	Automatic recovery of MCLK errors	
		0	Switches the MCLK to the ROSC when a clock error occurs.
		1	Disables the automatic recovery of the MCLK.
7	MCLKMNT	Whether to enable or disable MCLK monitoring	
		0	Disables MCLK monitoring
		1	Enables MCLK monitoring
6	MCLKIE	Whether to enable or disable the MCLK error interrupt	
		0	Disables the MCLK error interrupt.
		1	Enables the MCLK error interrupt.
5	MCLKFAIL	MCLK error interrupt flag	
		0	The MCLK error interrupt has not occurred.
		1	Read: The MCLK error interrupt has occurred. Write: Clears the interrupt flag.
4	MCLKSTS	MCLK clock status flag	
		0	The MCLK is oscillating normally.
		1	The MCLK is not oscillating.
3	HSEMNT	Whether to enable or disable external HSE monitoring	
		0	Disables external HSE monitoring.
		1	Enables external HSE monitoring.
2	HSEIE	Whether to enable or disable the external HSE error interrupt	
		0	Disables the external HSE error interrupt.
		1	Enables the external HSE error interrupt.
1	HSEFAIL	External HSE error interrupt flag	
		0	The external HSE error interrupt has not occurred.
		1	Read: The external HSE error interrupt has occurred. Write: Clears the interrupt flag.
0	HSESTS	External HSE status flag	
		0	The external HSE is oscillating normally.
		1	The external HSE is not oscillating.

NOTES:

1. Clock monitoring can be active only when the corresponding clock is enabled.
2. As clock monitoring is performed on the basis of the LSI, only those clocks that are faster than the LSI can be monitored.

4.5.19 SCU_COR: clock output register

The SCU_COR register defines the frequency division ratio at which the MCLK is fed to external devices. To use the CLKO pin in output mode, you must configure the pin mux. It is a 32-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																								CLKOINSEL			CLKOEN	CLKODIV			
-																								000			0	1111			
-																								RW			RW	RW			

7 5	CLKOINSEL	Selection of the clock to send a signal from CLKO	
		000	LSI
		100	MCLK
		101	HSI
		110	HSE
		111	PLL
		Other	Reserved
4	CLKOEN	Whether to enable or disable clock output from CLKO	
		0	Disables output from CLKO (maintains the “low” signal output).
		1	Enables output from CLKO.
3 0	CLKODIV	Clock output divider value CLKO = MCLK (CLKODIV = 0) $\mathrm{CLKO} = \frac{MCLK}{(\mathrm{CLKODIV} * 2)} \quad (CLKODIV > 0)$	

4.5.20 SCU_NMICR: NMI control register

The SCU_NMICR register controls NMIs (non-maskable interrupts), which are software-configurable. Any interrupt can be an NMI source. Once an NMI event occurs, it jumps to the NMI handler.

SCU_NMICR=0x4000_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	NMISRC	NIMINEN	Reserved	PROT1EN	OVP1EN	PROT0EN	OVP0EN	WDTEN	MCLKFAILEN	LVIEN
-	0x00	0	-	0	0	0	0	0	0	0
-	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

23 16	NMISRC	NMI source selection	
		One of the interrupt priority levels can be used as the NMI source.	
15	NMIINEN	Whether to enable or disable the NMIs (interrupts that have been specified as NMI sources)	
		0	Disables.
		1	Enables.
6	PROT1EN	Whether to enable or disable the MPWM1 protection interrupt as an NMI	
		0	Disables.
		1	Enables.
5	OVP1EN	Whether to enable or disable the MPWM1 overvoltage protection interrupt as an NMI	
		0	Disables.
		1	Enables.
4	PROT0EN	Whether to enable or disable the MPWM0 protection interrupt as an NMI	
		0	Disables.
		1	Enables.
3	OVP0EN	Whether to enable or disable the MPWM0 overvoltage protection interrupt as an NMI	
		0	Disables.
		1	Enables.
2	WDTINTEN	Whether to enable or disable the WDT interrupt as an NMI	
		0	Disables.
		1	Enables.
1	MCLKFAILEN	Whether to enable or disable the MCLK error interrupt as an NMI	
		0	Disables.
		1	Enables.
0	LVIEN	Whether to enable or disable the LVI error interrupt as an NMI	
		0	Disables.
		1	Enables.

4.5.21 SCU_NMISR: NMI status register

The SCU_NMISR register is used to clear NMI occurrence flags. Individual interrupt flags can be cleared after writing a specific key value to the identification key bit field.

SCU_NMISR=0x4000_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
WTIDKY	Reserved	NMIINTSTS	Reserved	PROT1STS	OVP1STS	PROT0STS	OVP0STS	WDTSTS	MCLKFAILSTS	LVISTS
0x00	-	0	-	0	0	0	0	0	0	0
WO	-	RO	-	RC	RC	RC	RC	RC	RC	RC

31 24	WTIDKY	Write identification key	
		Writing 0x8C to the bit field enables clearing flags in this register.	
15	NMIINTSTS	Whether or not an NMI has occurred (The flag can be cleared by writing to the triggered NMI's bit.)	
		0	No NMI has occurred.
		1	An NMI has occurred.
6	PROT1STS	Whether or not the MPWM1 protection interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
5	OVP1STS	Whether or not the MPWM1 overvoltage protection interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
4	PROT0STS	Whether or not the MPWM0 protection interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
3	OVP0STS	Whether or not the MPWM0 overvoltage protection interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
2	WDTINTSTS	Whether or not the WDT interrupt has occurred (To ensure that the bit is flagged only once, the timer must be reloaded before the flag is cleared.)	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
1	MCLKFAILSTS	Whether or not the MCLK error interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
0	LVISTS	Whether or not the LVI interrupt has occurred	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).

4.5.22 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	CTRLOTP	PLLRSTB	PLLEN	BYPASSB	PLLMODE	Reserved	PREDIV	POSTRIV1	POSTDIV2	OUTDIV
0	-	0110	0	0	0	0	-	000	00000000	0000	0000
RO	-	RW	RW	RW	RW	RW	-	RW	RW	RW	RW

31	PLLLOCK	PLLLOCK status			
		0	PLL is not locked		
		1	PLL is locked		
27 24	CTRLOTP	PLL current option			
		[27:26]	Current option	[25:24]	VCO bias
		00	5 uA	00	x 1/4
		01	10 uA	01	x 1/2
		10	15 uA	10	x 1
		11	20 uA	11	x 2
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		
20	PLLMODE	PLL voltage-controlled oscillator (VCO) mode selection			
		0	The VCO frequency is the same as FOUT.		
		1	The VCO frequency is twice FOUT.		
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 1 (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:

1. First, set the PLLEN bit to '1', and then set the PLLSTB and other bits after 1us or more.
2. Wait more than 190us for PLL Lock Time, then check the PLL Lock flag(Bit 31).
3. Output calculation formula is as followings:

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

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

4.5.23 SCU_VDCCON: VDC control register

The SCU_VDCCON register controls the VDC, which represents the MCU’s internal voltage. The VDCWDLY bit field defines the length of the SCU’s warm-up delay. This register is 32 bits wide.

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				VDCWDLY
-				0x7F
-				RW

7 0	VDCWDLY	VDC warm-up delay count value When the SCU is waked up from stop mode, the warm-up delay is inserted to wait until the VDC output is stabilized. The amount of delay can be set with this register to a maximum of 0x7F (2 msec).
NOTE: Reserved bits should never be modified.		

4.5.24 SCU_LVICR: LVI (Low-Voltage Indicator) control register

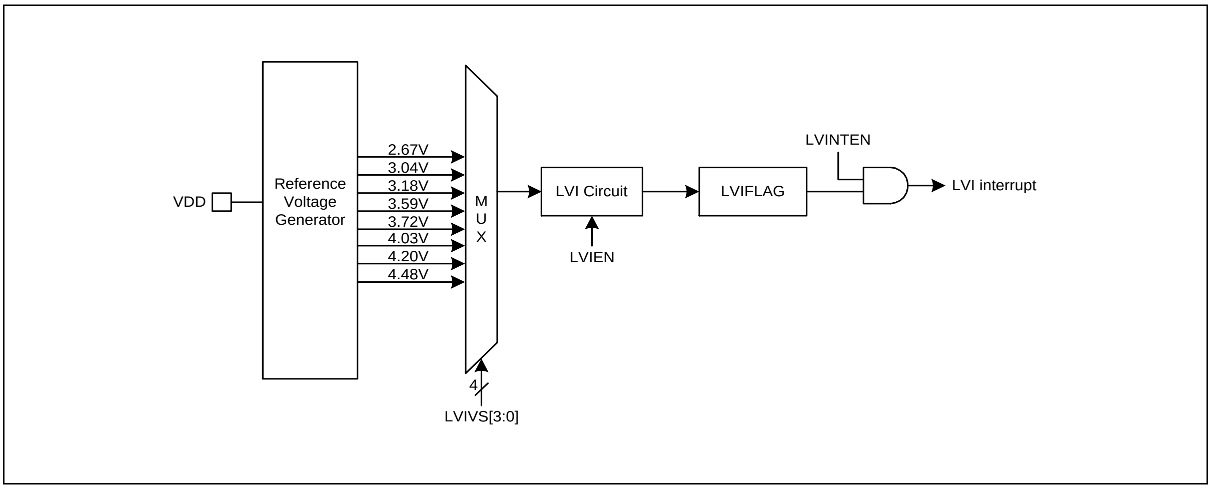
The SCU_LVICR register is 32 bits wide. It is used to control the LVI.

SCU_LVICR=0x4000_0068

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	LVIINTEN	LVIAON	LVIVS
-	0	-	0	0	0000
-	RW	-	RW	RW	RW

7	LVIEN	Whether to enable or disable the LVI	
		0	Disables.
		1	Enables.
5	LVIINTEN	Whether to enable or disable the LVI interrupt	
		0	Disables.
		1	Enables.
4	LVIAON	Whether or not to deactivate the LVI in stop mode	
		0	Disables LVI auto-off.
		1	Enables the LVI to automatically turn off when the MCU enters stop mode.
3 0	LVIVS	LVI voltage selection	
		1000	2.67V
		1001	3.04V
		1010	3.18V
		1011	3.59V
		1100	3.72V
		1101	4.03V
		1110	4.20V
		1111	4.48V

Figure 22. LVI Block Diagram



4.5.25 SCU_LVISR: LVI (Low-Voltage Indicator) status register

The SCU_LVISR register is 32 bits wide and indicates the LVI's operating status.

SCU_LVISR=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
WTIDKY	Reserved	LVIIFLAG	Reserved	LVIINTSTS
0x00	-	0	-	0
WO	-	RC	-	RO

31 24	WTIDKY	Write Identification Key
		Writing 0x7A to the bit field enables clearing flags in this register.
5	LVIIFLAG	LVI interrupt flag
		0 Not flagged.
		1 Flagged (Writing a 1 to the bit clears it to 0).
0	LVIINTSTS	Whether or not the LVI interrupt is active (raw data)
		0 The LVI interrupt has ended.
		1 The LVI interrupt is active.

4.5.26 SCU_LVRCR: LVR (Low-Voltage Reset) control register

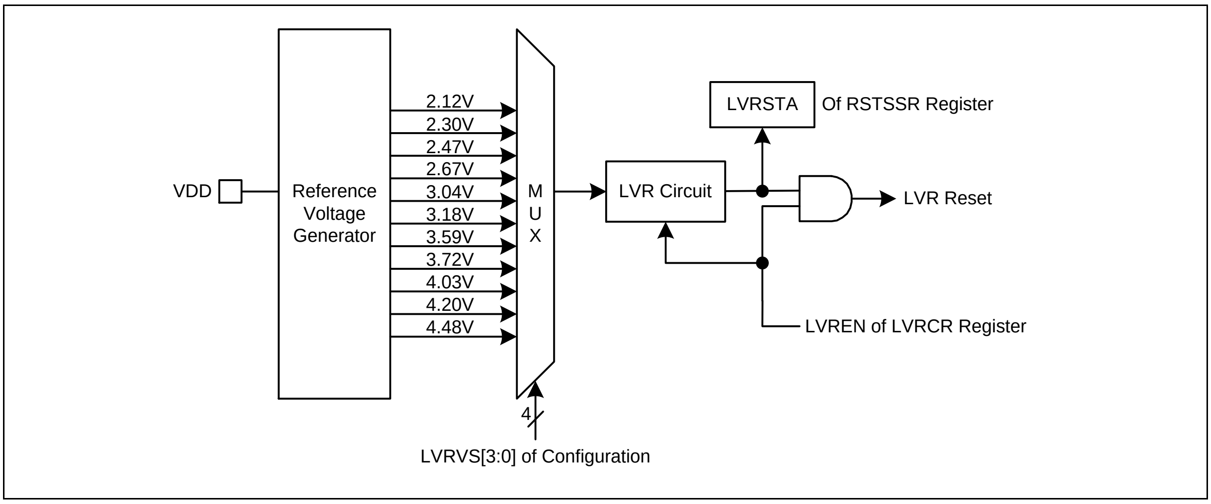
The SCU_LVRCR register is 32 bits wide and defines the default mode for LVR operation.

SCU_LVRCR=0x4000_0070

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	Reserved	LVRAON	LVRVS
-		0x00	-	0	0101
-		RW	-	RW	RW

15 8	LVREN	LVR reset operation control master configuration	
		0xAA	Disables LVR operation.
		Others	Enables LVR operation.
4	LVRAON	Whether or not to deactivate the LVR in stop mode	
		0	Disables LVR auto-off.
		1	Enables the LVR to automatically turn off when the MCU enters stop mode.
3 0	LVRVS	LVR reset level	
		0101	2.12V
		0110	2.30V
		0111	2.47V
		1000	2.67V
		1001	3.04V
		1010	3.18V
		1011	3.59V
		1100	3.72V
		1101	4.03V
		1110	4.20V
		1111	4.48V

Figure 23. LVR Block Diagram



4.5.27 SCU_EOSCR: external oscillator control register

The SCU_EOSCR register defines the drive current of the external main crystal oscillator and the delay time to reduce noises. It is a 32-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	HSEISEL	Reserved	HSENFEN	Reserved	HSENFSEL
-	00	-	0	-	00
-	RW	-	RW	-	RW

9 8	HSEISEL	MOSC current capability selection	
		00	$12M < f_{OUT} \leq 16M$
		01	$8M < f_{OUT} \leq 12M$
		10	$4M < f_{OUT} \leq 8M$
		11	$1M < f_{OUT} \leq 4M$
4	HSENFEN	HSE noise filter enablement	
		0	Disable.
		1	Enable.
0	HSENFSEL	HSE noise filter selection	
		00	23ns
		01	18ns
		10	13ns
		11	8ns

Table 18. External Oscillator Control Settings

Freq.(MHz)	HSENFSEL [1:0]	HSEISEL[1:0]	NC Delay(ns)
4	[00b]	[11b]	23ns
8	[01b]	[10b]	18ns
12	[10b]	[01b]	13ns
16	[11b]	[00b]	8ns

4.5.28 SCU_MCCR1: miscellaneous clock control register 1

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

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	WDTCSEL	WDTCDIV	Reserved	STCSEL	STCDIV
-	000	0x01	-	000	0x00
-	RW	RW	-	RW	RW

26 24	WDTCSEL	WDT clock source selection	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	WDTCDIV	WDT clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	
10 8	STCSEL	SYSTICK Clock source select bit	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	STDIV	SYSTICK clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	

4.5.29 SCU_MCCR2: miscellaneous clock control register 2

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

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	MPWM1CSEL	MPWM1CDIV	Reserved	MPWM0CSEL	MPWM0CDIV
-	000	0x00	-	000	0x00
-	RW	RW	-	RW	RW

26 24	MPWM1CSEL	MPWM1 clock source selection	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	MPWM1CDIV	MPWM1 clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	
10 8	MPWM0CSEL	MPWM0 Clock source select bit	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	MPWM0CDIV	MPWM0 clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	

4.5.30 SCU_MCCR3: miscellaneous clock control register 3

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

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	TIMER47CSEL	TIMER47CDIV	Reserved	TIMER03CSEL	TIMER03CDIV
-	000	0x00	-	000	0x00
-	RW	RW	-	RW	RW

Bit	Name	Description	
26 24	TIMER47CSEL	Clock source selection for timer 4 through timer 7	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	TIMER47CDIV	Clock divider N for timer 4 through timer 7 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0. If the timer clock (TCLK) is set in this register, the clock speed must satisfy PCLK ≥ MCCR x 2.	
10 8	TIMER03CSEL	Clock source selection for timer 0 through timer 3	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	TIMER03CDIV	Clock divider N for timer 0 through timer 3 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0. If the timer clock (TCLK) is set in this register, the clock speed must satisfy PCLK ≥ MCCR x 2.	

NOTE: Because PCLK is the standard of operation, MCCR value should be set slower than PCLK. Therefore, if MCCR value uses the same frequency as PCLK, it should be divided more than 2 times. If PCLK is 1/2 slower than MCCR clock, frequency should be divided more than 4 times.

4.5.31 SCU_MCCR4: miscellaneous clock control register 4

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

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	PGADCSEL	PGADCDIV	Reserved	ADCCSEL	ADCCDIV
-	000	0x00	-	000	0x00
-	RW	RW	-	RW	RW

26 24	PGADCSEL(PA,PB)	Clock source for port group A debouncing	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	PGADCDIV(PA,PB)	Clock divider N for port group A debouncing 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	
NOTE: Clock is not activated during the PDEx bit in the Pn_DER register is disabled.			
10 8	ADCCSEL	ADC clock source selection	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	ADCCDIV	ADC clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 2, and the divided clock must be slower than the system clock.	
NOTE: ADC clock's frequency does not exceed 25MHz.			

4.5.32 SCU_MCCR5: miscellaneous clock control register 5

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

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	PGCDCSEL	PGCDCDIV	Reserved	PGBDCSEL	PGBDCDIV
-	000	0x00	-	000	0x00
-	RW	RW	-	RW	RW

26 24	PGCDCSEL(PF,PG)	Clock source for port group C debouncing	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	PGCDCDIV(PF,PG)	Clock divider N for port group C debouncing 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	
NOTE: Clock is not activated during the PDEx bit in the Pn_DER register is disabled.			
10 8	PGBDCSEL(PC,PD)	Clock source for port group B debouncing	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	PGBDCDIV(PC,PD)	Clock divider N for port group B debouncing 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	
NOTE: Clock is not activated during the PDEx bit in the Pn_DER register is disabled.			

4.5.33 SCU_MCCR6: miscellaneous clock control register 6

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

SCU_MCCR6=0x4000_00A4

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">Reserved</td><td colspan="3">FRTCSEL</td><td colspan="8">FRTCDIV</td></tr>
<tr><td colspan="21">-</td><td colspan="3">000</td><td colspan="8">0x00</td></tr>
<tr><td colspan="21">-</td><td colspan="3">RW</td><td colspan="8">RW</td></tr>
</table>

10 8	FRTCSEL	FRT clock source selection	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
7 0	FRTCDIV	FRT clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0.	

4.5.34 SCU_MCCR7: miscellaneous clock control register 7

Based on the settings of the SCU_MCCR registers, the A33M11x series allows for a different MCLK division ratio for each peripheral. This register is 32 bits wide.

SCU_MCCR7=0x4000_00A8

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	UARTCSEL	UARTCDIV	Reserved
-	000	0x01	-
-	RW	RW	-

26 24	UARTCSEL	UART clock source selection (UART0 through UART3)	
		000	LSI(500kHz)
		100	MCLK
		101	HSI(32MHz)
		110	HSE
		111	PLL
		Other	Reserved
23 16	UARTCDIV	UART clock divider N 0x00: Disables the divider. 0xN: Clock source / N N must be larger than 0. If the UART clock is set in this register, the clock speed must satisfy PCLK ≥ MCCR x 2.	

4.5.35 SCU_SYSTEN: system access enable register

The SCU_SYSTEN register determines whether or not to allow changes to the settings of all SCU registers.

SCU_SYSTEN=0x4000_00F0

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			ENS	SYSTEN
-			0	--
-			RO	WO

8	ENS		Whether the register is enabled or disabled
		0	Disable SYSTEN.
		1	Enable SYSTEN (SCU registers can be accessed).
7 0	SYSTEN		Writing 0x57 and then 0x75 to the bit field enables writing new values to SCU registers. After this, write a different value to this bit field to protect the SCU registers against being updated with new values. However, access to the SCU_NMISR and SCU_LVISR registers is not defined by the SCU_SYSTEM register. Their bits can be cleared regardless of SCU_SYSTEN enablement.

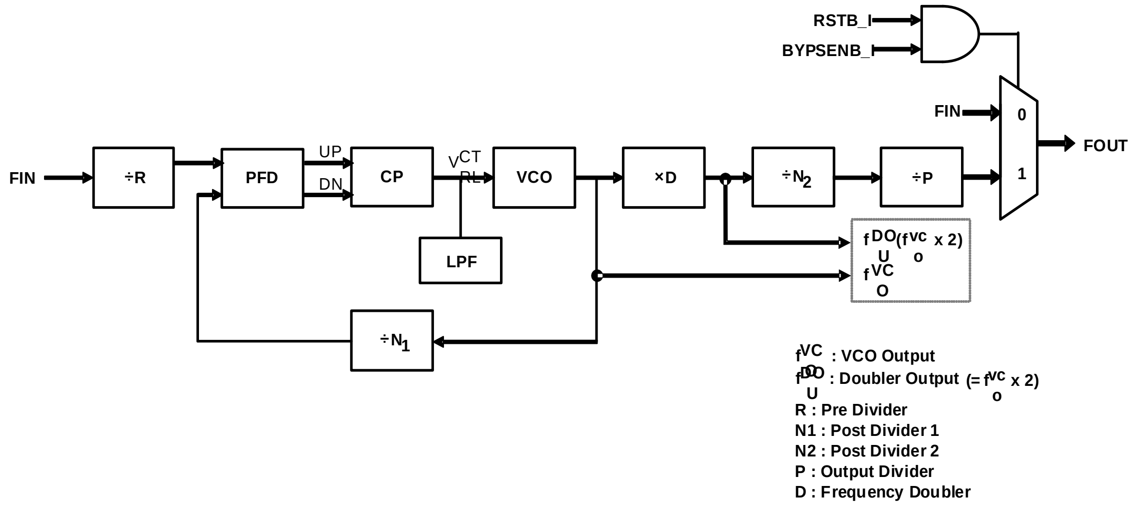
NOTE: Example code for using SYSTEN

```
        SYSTEN=0x57; SYSTEN=0x75    // Enables SYSTEN.
                                    // SCCR, CSCR, PLLCON, etc. become settable
        SYSTEN=0;                   // Disables SYSTEN.
```

4.6 Functional description

4.6.1 PLL clock setting

Figure 24. PLL Block Diagram



NOTE: FIN is the clock before pre dividing.

For the A33M11x series, the PLL's output frequency (FOUT) can be precisely set in 1MHz steps. The formula below calculates the input frequency (FIN) fed to the PLL's VCO. Although a frequency of 1–3MHz is supported, it is recommended to set FIN to 2MHz if possible.

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

The VCO's FOUT must be set between 50 and 200MHz and calculated by the formula below:

$$\mathrm{VCO} = \mathrm{FIN} \times (\mathrm{N}_1 + 1), \quad 50\mathrm{MHz} \le \mathrm{VCO} \le 200\mathrm{MHz}\ \mathit{if}\ \mathrm{D} = 0$$

SCU_PLLCON provides a "doubler" that doubles the VCO output based on the setting of the PLLMODE bit. When the doubler is used, the VCOx2 output must be no larger than 250MHz. The formula below calculates VCOx2:

$$\mathrm{VCOx2} = \mathrm{VCO} \times (\mathrm{D} + 1), \qquad 100\mathrm{MHz} \le \mathrm{VCOx2} \le 250\mathrm{MHz}\ \mathit{if}\ \mathrm{D} = 1$$

Based on the formulas above, the final FOUT can be calculated by the following formula:

$$\mathrm{FOUT} \ = \frac{\mathrm{PLLINCLK} \times (\mathrm{N}_1 + 1)}{(\mathrm{R}+1) \times (\mathrm{N}_2 + 1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1) \ = \frac{\mathrm{FIN} \times (\mathrm{N}_1 + 1)}{(\mathrm{N}_2 + 1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1)$$

4.6.2 Clock monitoring

If clock monitoring is enabled, a reset or an interrupt occurs when the oscillator stops operating due to external noise or other causes. In this case, the core clock must not be the HSE. Because the internal LSI is used to monitor clocks, this oscillator must be set to operate at all times.

The following figure illustrates HSE operation and clock monitoring.

Figure 25. HSE Operation Workflow

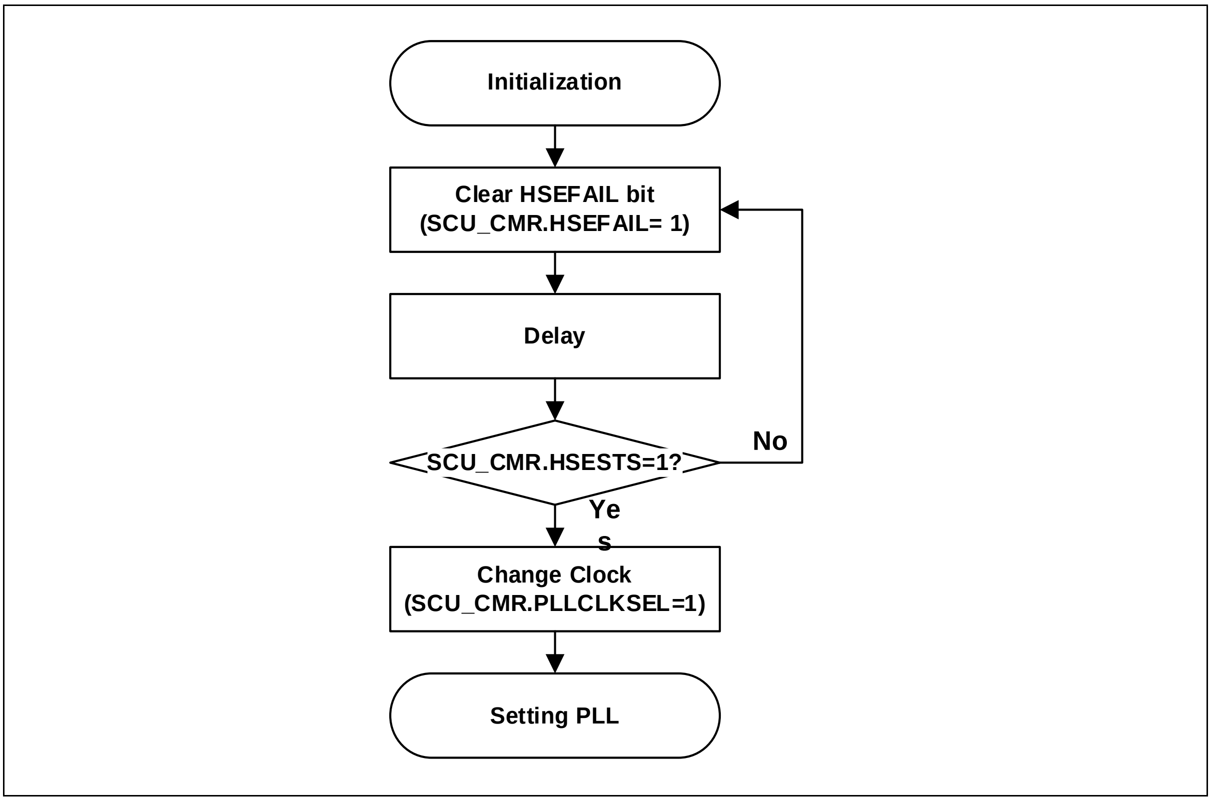
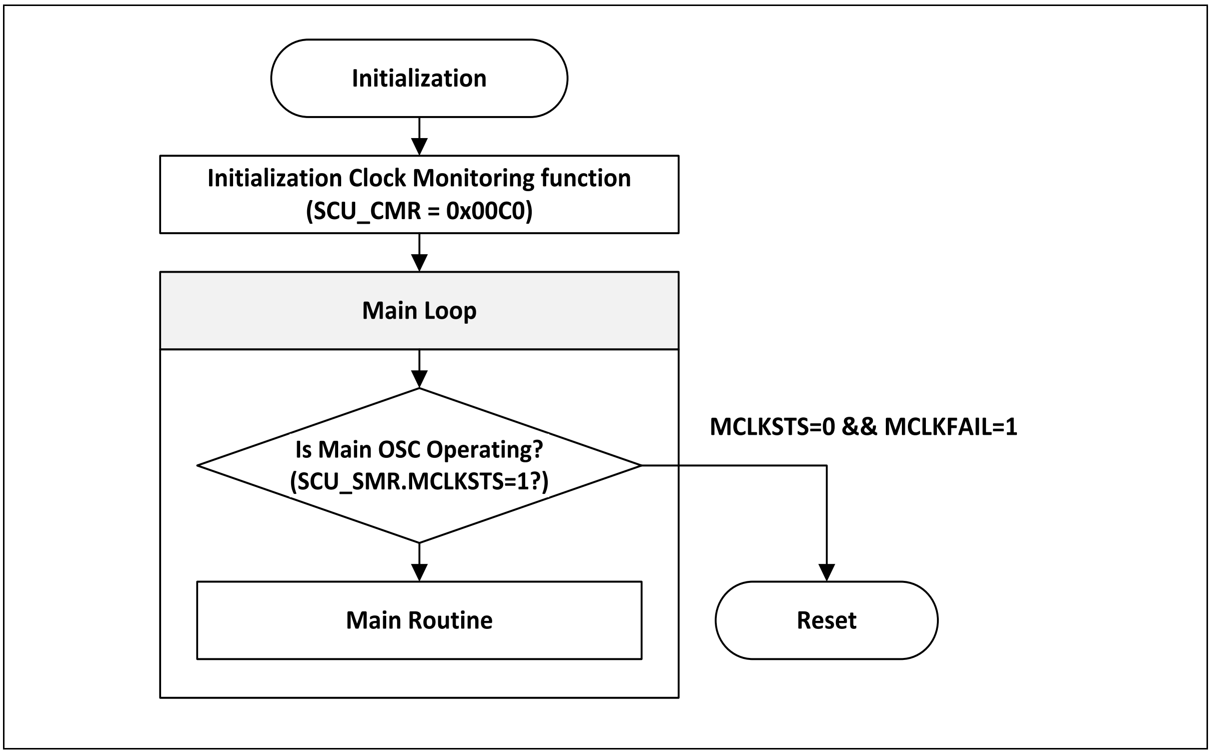


Figure 26. Clock Monitoring Workflow



4.6.3 Setting examples

<Example 1> Setting the LSI as the system clock

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”                            : Unlocks the SCU registers
SCU_SYSTEN<STSTEN[7:0]> = “0x75”
SCU_CSCR = “0xAA”                                           : Enables all clocks.

Stable Time;                                                : Stabilization time

SCU_SCCR<HCLKDIV[27:24]> = “0x0”                            : HCLK = MCLK / 1
SCU_SCCR<PCLKDIV[18:16]> = “0x0”                            : PCLK = HCLK / 1
SCU_SCCR<MCLKSEL[2:0]> = “0x0”                              : MCLK = LSI
SCU_SYSTEN = “0”                                            : Locks the SCU registers.
```

<Example 2> Enabling HSE monitoring with the HSE as the system clock

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                            : Unlocks the SCU registers.
SCU_CSCR = “0x22”                                           : Enables the LSI and HSE.
SCU_CMR<HSEMNT[3]> = “1”                                    : Enables HSE monitoring.
SCU_CMR<HSEFAIL[1]> = “1”                                   : Clears the HSE monitoring flag.
SCU_SYSTEN = “0”                                            : Locks the SCU registers.
```

<Example 3> Setting the PLL frequency (to 96MHz)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                            : Unlocks the SCU registers.
SCU_SCCR<PLLCLKSEL[12]> = “1”                               : Sets the HSE as the PLL input clock.
SCU_PLLCON<PLLRSTB[23]> = “1”                               : Enables the PLL reset.
SCU_PLLCON<PLLEN[22]> = “1”                                 : Enables the PLL.
SCU_PLLCON<BYPASSB[21]> = “1”                               : Selects the PLL output.
SCU_PLLCON<PLLMODE[20]> = “0                                : Sets the PLL output to FOUT x1.
SCU_PLLCON<PREDIV[18:16]> = “011”                           : Sets PREDIV to 3.
SCU_PLLCON<POSTDIV1[15:8]> = “0x2F”                         : Sets POSTDIV1 to 47.
SCU_PLLCON<POSTDIV2[7:4]> = “000”                           : Sets POSTDIV2 to 0.
SCU_PLLCON<OUTDIV[3:0]> = “000”                             : Sets OUTDIV to 0.
SCU_PLLCON<LOCKSTS[31]> Lock Check                          : Checks for PLL lock.
SCU_SCCR<MCLKSEL[2:0]> = “111”                              : MCLK = PLL
SCU_SYSTEN = “0”                                            : Locks the SCU registers.
```

5 Port Control Unit (PCU) and GPIO

Port Control Unit (PCU)

A33M11x MCU's Port Control Unit (PCU) block controls the external input and output (I/O) ports. By setting the PCU block registers, you can configure the pins uses, input/output, pull-up/pull-down, and debouncing, as needed for your application.

The PCU block configures and controls external I/Os as follows:

- The MUX registers define the use of each pin.
  - Input/output
  - Push-pull output
  - Open-drain output
  - Logic input
  - Analog input
- The internal pull-up resistor and open-drain mode are configurable for each pin.
- The following interrupts can be set for each pin:
  - Input level interrupt
  - Input rising-edge interrupt
  - Input falling-edge interrupt
  - Input both-edge interrupt
- Up to six GPIO interrupts are supported (GPIOA through GPIOG).
- Each pin can be set for debouncing.

General Purpose Input/Output (GPIO)

Pins other than the VDD, GND, and certain specific-purpose pins can be used as General Purpose Input/Output (GPIO) pins.

The GPIO block controls the general I/O ports. Output pins can be configured by setting their bits to generate an “H” or “L” level signal, and logic input pins can be checked for their input state.

GPIO ports are controlled by GPIO block as listed in the followings:

- Output signal level (H/L) selection
- External interrupt interface
- Enables or disables pull-up/pull-down for pins

Six pins in Table 19 are assigned for PCU and GPIO blocks.

Table 19. PCU and GPIO Pins

Pin name	Type	Description
PA	IO	PA0 to PA15
PB	IO	PB0 to PB15
PC	IO	PC0 to PC15
PD	IO	PD0 to PD3
PF	IO	PF0 to PF4, PF6 to PF7
PG	IO	PG10

5.1 PCU and GPIO block diagram

Figure 27 describes PCU in block diagram.

Figure 27. PCU Block Diagram

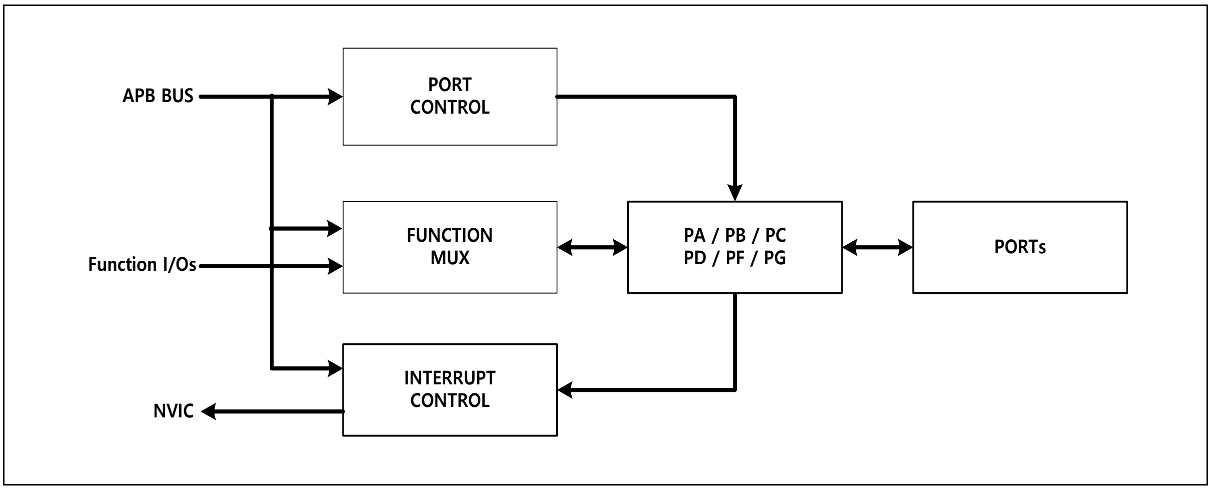


Figure 28 describes GPIO in block diagram, and Figure 29 introduces external interrupt I/O pins.

Figure 28. GPIO Block Diagram

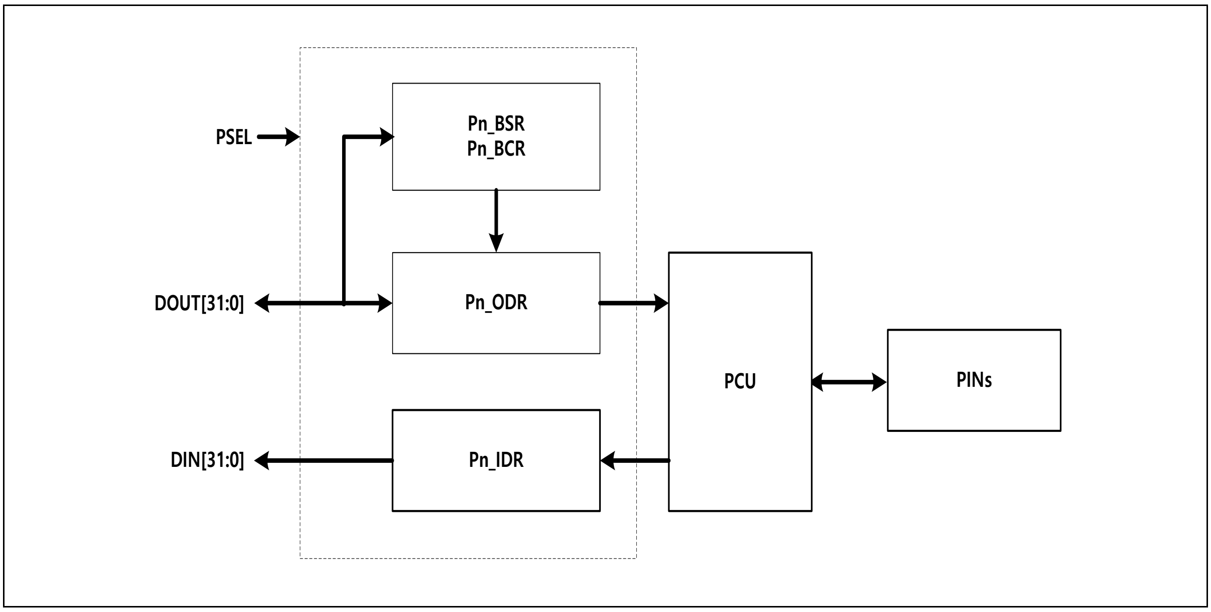


Figure 29. I/O Port Block Diagram: GPIO Pins

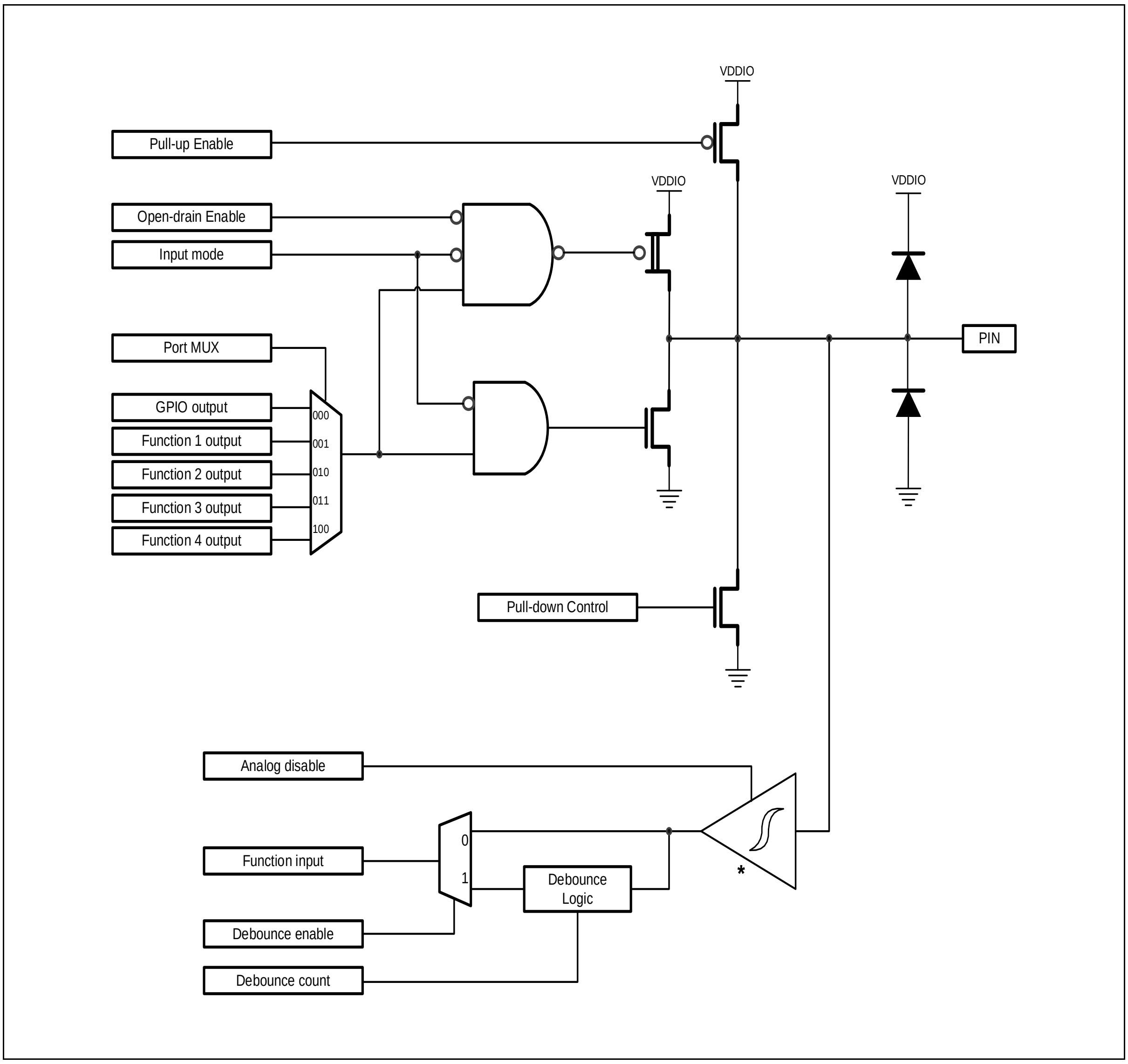
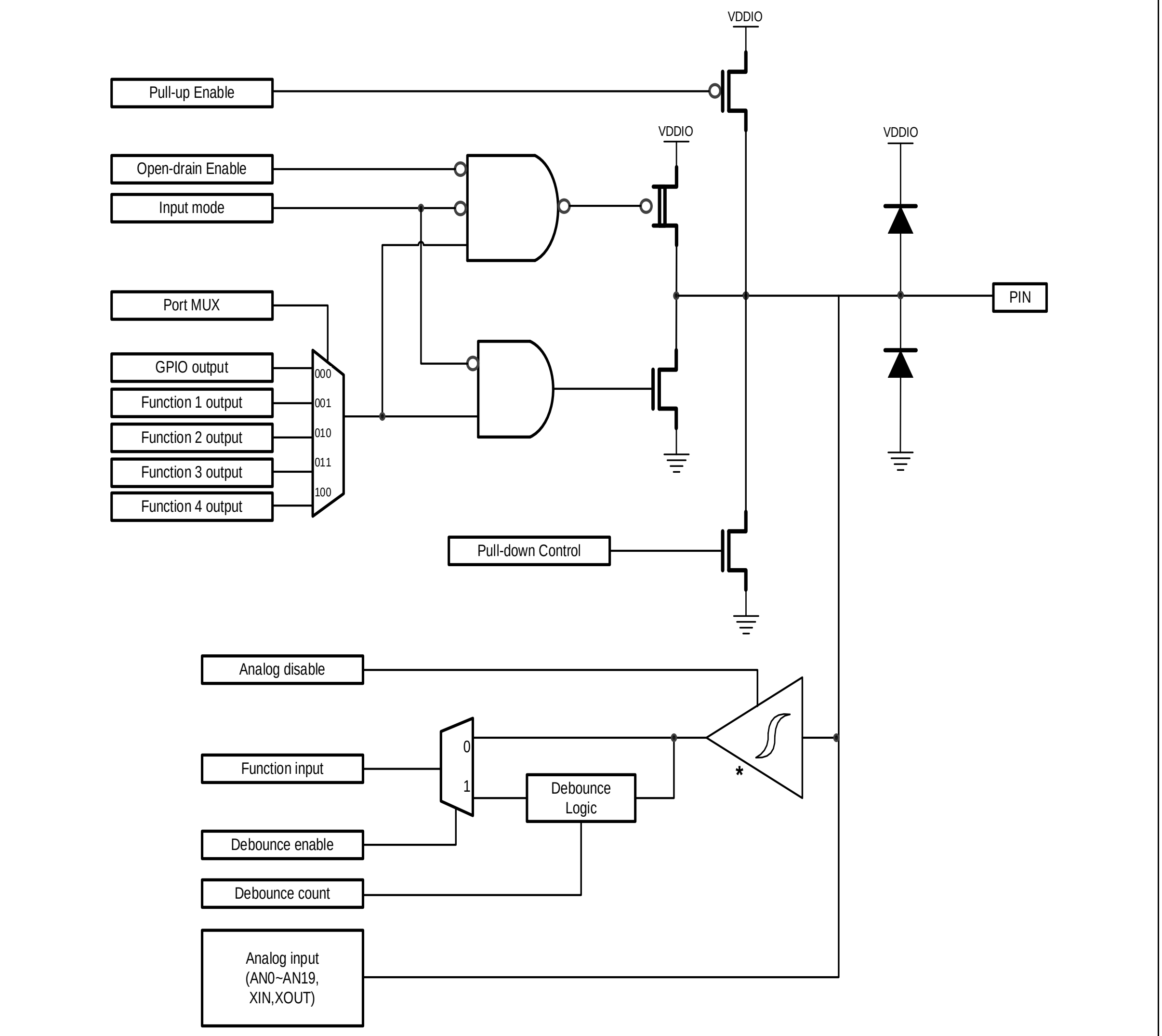


Figure 30. I/O Port Block Diagram: ADC and External Oscillator Pins



5.2 Pin multiplexing

GPIO pins have alternative function pins. Table 20 shows pin multiplexing information.

Table 20. GPIO Alternative Function

Pin name	Alternative function					
	AF0	AF1	AF2	AF3	AF4	AF7
PA0	PA0*				CO0	AN0/AO0
PA1	PA1*					AN1/AIN0
PA2	PA2*					AN2/AIP0
PA3	PA3*			CAPEU		AN3/AIP1
PA4	PA4*		T0IO			AN4
PA5	PA5*		T1IO			AN5
PA6	PA6*		T2IO	CAPEV		AN6/AIN1
PA7	PA7*		T3IO	CAPEW	CO1	AN7/AO1
PA8	PA8*			SCAPEU	CO2	AN8/AO2
PA9	PA9*			SCAPEV		AN9/AIN2
PA10	PA10*		QEI0_UPDN	SCAPEW		AN10/AIP2
PA11	PA11					AN11
PA12	PA12	SS0				AN12
PA13	PA13*	SCK0	QEI0_A			AN13/AIP3
PA14	PA14*	MOSI0	QEI0_B	PRTINEV		AN14/AIN3
PA15	PA15*	MISO0	QEI0_IDX	OVINEV	CO3	AN15/AO3
PB0	PB0*			MP0UH		
PB1	PB1*			MP0UL		
PB2	PB2*			MP0VH		
PB3	PB3*			MP0VL		
PB4	PB4*		T4IO	MP0WH		
PB5	PB5*		T5IO	MP0WL		
PB6	PB6*			PRTIN0U		
PB7	PB7*			OVIN0U	CLKO	
PB8	PB8*	RXD3		PRTIN1U	SWCLK	
PB9	PB9*	TXD3		OVIN1U	SWDIO	
PB10	PB10*		SS1	MP1UH		
PB11	PB11*		SCK1	MP1UL		
PB12	PB12*	TXD2	MOSI1	MP1VH		
PB13	PB13*	RXD2	MISO1	MP1VL		
PB14	PB14*		PRTINEV	MP1WH		
PB15	PB15*		OVINEV	MP1WL		

Table 20. GPIO Alternative Function (continued)

Pin name	Alternative function					
	AF0	AF1	AF2	AF3	AF4	AF7
PC0	PC0	RXD0		SWCLK*	nBOOT	
PC1	PC1	TXD0		SWDIO*		
PC2	PC2*					
PC3	PC3*					
PC4	PC4*		T0IO			
PC5	PC5*	RXD1	T1IO	QEI1_UPDN		
PC6	PC6*	TXD1	T2IO	QEI1_A		
PC7	PC7*	SCL0	T3IO	QEI1_B		
PC8	PC8*	SDA0	T4IO	QEI1_IDX		
PC9	PC9*		T5IO	CLKO		
PC10	PC10			nRESET*		
PC11	PC11		T6IO	nBOOT*		
PC12	PC12*					XIN
PC13	PC13*					XOUT
PC14	PC14*			RXD0		
PC15	PC15*			TXD0		
PD0	PD0*	SS0	T6IO			AN16
PD1	PD1*	SCK0	T7IO			AN17
PD2	PD2*	MOSI0	SCL1			AN18
PD3	PD3*	MISO0	SDA1			AN19
PF0	PF0*	SCL1				
PF1	PF1*	SDA1				
PF2	PF2*					
PF3	PF3*					
PF4	PF4*					
PF5	PF6*			PRTINEW		
PF6	PF7*			OVINEW		
PG10	PG10*		T7IO			

NOTES:

1. The ‘*’ means ‘selected pin function after reset condition’, which is a 64-pin standard.(The initial value of the pin is different depending on the package type)
2. Unused pins are set to output from Firmware (low output is recommended).

5.3 Registers

Base address of PCU is introduced in the followings:

Table 21. 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
PD	0x4000_1300	General Port D
PF	0x4000_1500	General Port F
PG	0x4000_1600	General Port G

Table 22. PCU and GPIO Register Map

Name	Offset	Type	Description	Reset value	Reference
Pn_MR1	0x0000	RW	Port n MUX1 select register	PA:(64Pin)0x7777_7777/ (48/44Pin)0x7777_7777 PB:(64Pin)0x7777_7777/ (48/44Pin)0x7777_7777 PC:(64Pin)0x7777_7733/ (48/44Pin)0x7777_7704 PD:(64Pin)0x0000_7777/ (48/44Pin)0x0000_7777 PF:(64Pin)0x7707_7777/ (48/44Pin)0x7707_7777 PG:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000	5.3.1
Pn_MR2	0x0004	RW	Port n MUX2 select register	PA:(64Pin)0x7777_7777/ (48/44Pin)0x7777_7777 PB:(64Pin)0x7777_7700/ (48/44Pin)0x7777_7744 PC:(64Pin)0x7077_3377/ (48/44Pin)0x7077_0377 PD:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000 PF:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000 PG:(64Pin)0x0000_0700/ (48/44Pin)0x0000_0700	5.3.2
Pn_CR	0x0008	RW	Port n control register	PA:(64Pin)0xFFFF_FFFF/ (48/44Pin)0xFFFF_FFFF PB:(64Pin)0xFFFF_FFFF/ (48/44Pin)0xFFFF_FFFF PC:(64Pin)0xFFFF_FFFF/ (48/44Pin)0xFFFF_FFFF PD:(64Pin)0x0000_00FF/ (48/44Pin)0x0000_00FF PF:(64Pin)0x0000_F3FF/ (48/44Pin)0x0000_F3FF PG:(64Pin)0x0030_0000/ (48/44Pin)0x0030_0000	5.3.3
Pn_PRCR	0x000C	RW	Port n pull-up/ pull-down resistor control register	PA:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000 PB:(64Pin)0x000A_0000/ (48/44Pin)0x000A_0000 PC:(64Pin)0x20A0_000A/ (48/44Pin)0x20A0_000A PD:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000 PF:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000 PG:(64Pin)0x0000_0000/ (48/44Pin)0x0000_0000	5.3.4
Pn_DER	0x0010	RW	Port n debouncing enable register	0x0000_0000	5.3.5
Pn_STR	0x0014	RW	Port n strength configuration register	0x0000_0000	5.3.6

Table 22. PCU and GPIO Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
Pn_IER	0x0020	RW	Port n interrupt enable register	0x0000_0000	5.3.7
Pn_ISR	0x0024	RC	Port n interrupt status register	0x0000_0000	5.3.8
Pn_ICR	0x0028	RW	Port n interrupt control register	0x0000_0000	5.3.9
Pn_ODR	0x0030	RW	Port n output data register	0x0000_0000	5.3.10
Pn_IDR	0x0034	RO	Port n input data register	0x0000_0000	5.3.11
Pn_BSR	0x0038	WO	Port n set/reset register	0x0000_0000	5.3.12
Pn_BCR	0x003C	WO	Port n reset register	0x0000_0000	5.3.13
PORTEN	0x1FF0	RW	Port access enable register	0x0000_0000	5.3.14

NOTE: Where n = A, B, C, D, F and G.

5.3.1 Pn_MR1: PORT n mux1 select register

The Pn_MR1 register selects the mode for each of ports n 0–7. Before the ports are used, the register must be set correctly; otherwise, their proper operation is not ensured.

PA_MR1=0x4000_1000, PB_MR1=0x4000_1100, PC_MR1=0x4000_1200
PD_MR1=0x4000_1300, PF_MR1=0x4000_1500, PG_MR1=0x4000_1600

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	P7MUX	Reserved	P6MUX	Reserved	P5MUX	Reserved	P4MUX	Reserved	P3MUX	Reserved	P2MUX	Reserved	P1MUX	Reserved	P0MUX
-	111	-	111	-	111	-	111	-	111	-	111	-	111	-	111
-	RW	-	RW	-	RW	-	RW	-	RW	-	RW	-	RW	-	RW

4x+2 4x	PxMUX	Port muxing selection, x = 0–7	
		000	Function 0 (b'000)
		001	Function 1 (b'001)
		010	Function 2 (b'010)
		011	Function 3 (b'011)
		100	Function 4 (b'100)
		111	Function 7 (b'111)
		Others	Reserved

NOTES:

1. The PxMUX's PC12 and PC13 bits cannot be modified while the HSE is set as the system clock (MCLK).
2. If you select pin multiplexing for a pin, the corresponding I/O settings are automatically modified to meet your selection. (If Alternate Function is applied according to the Pn_MR register settings, I/O is automatically set as the input or output of the function.)
3. When pin multiplexing is set to AF1, AF2, … and AF7, it cannot be set to open-drain output or input even if users use the Pn_CR register.
4. When pin multiplexing is set to AF0, users can use the Pn_CR register to set a port.
5. When the MCU power is driven by 5V and UART or outside of Timer is connected to 3.3V to operate, the existing open-drains and pull-ups cannot be used.
6. Pn_MR1, Pn_MR2, Pn_CR, and Pn_PRCR are registers that can change the value when PORTEN is activated.
7. For the analog outputs, the PxMux bits of the Pn_MR1 and Pn_MR2 registers must be set to b'111 (Function 4) and the Px bit of the Pn_CR register must be set to b'1x (input).
   If the Px bit of the Pn_CR register is set to b'00 (push-pull output) or b'01 (open-drain output), the port may not operate correctly.
   Analog output pins are listed below:
   PA0(AO0), PA7(AO1), PA8(AO2), PA15(AO3), PC13(XOUT)
8. PC[8:7] are Open-drain pins. When using a port corresponding to these pins for Timer, I2C, QEI, etc., applications must use external Pull-up resistors.
9. Exceptionally, the reset values of the PC_MR1 / PD_MR1 / PF_MR1 / PG_MR1 registers are 0x7777_7733 (64PKG) or 0x7777_7704 (48/44PKG), 0x0000_7777, 0x7707_7777, and 0x0000_0000, respectively.

5.3.2 Pn_MR2: PORT n mux2 select register

The Pn_MR2 register selects the mode for each of ports n 8–15. Before the ports are used, the register must be set correctly; otherwise, their proper operation is not ensured.

PA_MR2=0x4000_1004, PB_MR2=0x4000_1104, PC_MR2=0x4000_1204
PD_MR2=0x4000_1304, PF_MR2=0x4000_1504, PG_MR2=0x4000_1604

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	P15MUX	Reserved	P14MUX	Reserved	P13MUX	Reserved	P12MUX	Reserved	P11MUX	Reserved	P10MUX	Reserved	P9MUX	Reserved	P8MUX
-	111	-	111	-	111	-	111	-	111	-	111	-	111	-	111
-	RW	-	RW	-	RW	-	RW	-	RW	-	RW	-	RW	-	RW

4(x-8)+2 4(x-8)	PxMUX	Port muxing selection, x = 8–15	
		000	Function 0 (b'000)
		001	Function 1 (b'001)
		010	Function 2 (b'010)
		011	Function 3 (b'011)
		100	Function 4 (b'100)
		111	Function 7 (b'111)
		Others	Reserved

NOTES:

1. The PxMUX's PC12 and PC13 bits cannot be modified while the HSE is set as the system clock (MCLK).
2. If you select pin multiplexing for a pin, the corresponding I/O settings are automatically modified to meet your selection. (If Alternate Function is applied according to the Pn_MR register settings, I/O is automatically set as the input or output of the function.)
3. When pin multiplexing is set to AF1, AF2, … and AF7, it cannot be set to open-drain output or input even if users use the Pn_CR register.
4. When pin multiplexing is set to AF0, users can use the Pn_CR register to set a port.
5. When the MCU power is driven by 5V and UART or outside of Timer is connected to 3.3V to operate, the existing open-drains and pull-ups cannot be used.
6. Pn_MR1, Pn_MR2, Pn_CR, and Pn_PRCR are registers that can change the value when PORTEN is activated.
7. For the analog outputs, the PxMux bits of the Pn_MR1 and Pn_MR2 registers must be set to b'111 (Function 4) and the Px bit of the Pn_CR register must be set to b'1x (input).
   If the Px bit of the Pn_CR register is set to b'00 (push-pull output) or b'01 (open-drain output), the port may not operate correctly.
   Analog output pins are listed below:
   PA0(AO0), PA7(AO1), PA8(AO2), PA15(AO3), PC13(XOUT)
8. PC[8:7] are Open-drain pins. When using a port corresponding to these pins for Timer, I2C, QEI, etc., applications must use external Pull-up resistors.
9. Exceptionally, the reset values of the PB_MR2 / PC_MR2 / PD_MR2 / PF_MR2 / PG_MR2 registers are 0x7777_7700 (64PKG) or 0x7777_7744 (48/44PKG), 0x7077_3377 (64PKG) or 0x7077_0377 (48/44PKG), 0x0000_0000, 0x0000_0000, and 0x0000_0700, respectively.

5.3.3 Pn_CR: PORT n control register

The Pn_CR register sets the pins in each port as input, open-drain output, or push-pull port pins.

PA_CR=0x4000_1008, PB_CR=0x4000_1108, PC_CR=0x4000_1208
PD_CR=0x4000_1308, PF_CR=0x4000_1508, PG_CR=0x4000_1608

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
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	Px	Port control, x = 0–15	
		00	Push-pull output
		01	Open-drain output
		1x	Input

NOTES:

1. When pin multiplexing is set to AF1 to AF7, it cannot be set even if the user sets it to open-drain output or input using the Pn_CR register.
2. When pin multiplexing is set to AF0, the Pn_CR register port setting is valid.
3. Pn_MR1, Pn_MR2, Pn_CR, and Pn_PRCR are registers that can change the value when PORTEN is activated.
4. For the analog outputs, the PxMux bits of the Pn_MR1 and Pn_MR2 registers must be set to b’111 (Function 4) and the Px bit of the Pn_CR register must be set to b’1x (input).
   If the Px bit of the Pn_CR register is set to b’00 (push-pull output) or b’01 (open-drain output), the port may not operate correctly.
   Analog output pins are listed below:
   PA0(AO0), PA7(AO1), PA8(AO2), PA15(AO3), PC13(XOUT)
5. Exceptionally, the reset values of the PD_CR, PF_CR, and PG_CR registers are 0x0000_00FF, 0x0000_F3FF, and 0x0030_0000, respectively.

5.3.4 Pn_PRCR: PORT n pull-up resistor control register

The Pn_PRCR register determines the enablement of the pull-up/pull-down resistors of each port pin

PA_PRCR=0x4000_100C, PB_PRCR=0x4000_110C, PC_PRCR=0x4000_120C
PD_PRCR=0x4000_130C, PF_PRCR=0x4000_150C, PG_PRCR=0x4000_160C

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
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
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	Px	Port pull-up/pull-down control, x = 0–15	
		0x	Disables the pull-up/pull-down resistors.
		10	Enables the pull-up resistor/disables the pull-down resistor.
		11	Disables the pull-up resistor/enables the pull-down resistor.

NOTES:

1. Pn_MR1, Pn_MR2, Pn_CR, and Pn_PRCR are registers that can change the value when PORTEN is activated.
2. Exceptionally, the reset values of the PB_PRCR / PC_PRCR registers are 0x000A_0000 and 0x20A0_000A, respectively.

5.3.5 Pn_DER: PORT n debouncing enable register

All pins in each port have a digital debouncing filter, which can be set in the Pn_DER register.

PA_DER=0x4000_1010, PB_DER=0x4000_1110, PC_DER=0x4000_1210
PD_DER=0x4000_1310, PF_DER=0x4000_1510, PG_DER=0x4000_1610

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	PDE15	PDE14	PDE13	PDE12	PDE11	PDE10	PDE9	PDE8	PDE7	PDE6	PDE5	PDE4	PDE3	PDE2	PDE1	PDE0
-	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	PDEx	Whether to enable or disable pin debouncing, x = 0–15	
		0	Disables the debouncing filter.
		1	Enables the debouncing filter.

NOTES:

1. To use the port debounce function, the debounce clock of SCU_MCCR4 and SCU_MCCR5 must be enabled first. If the debounce function is activated without setting the debounce clock, the port may malfunction.
2. Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.6 Pn_STR: PORT n strength configuration register

The Pn_STR register configures each pin’s drive strength. These settings affect the port’s speed and current consumption.

PA_STR=0x4000_1014, PB_STR=0x4000_1114, PC_STR=0x4000_1214
PD_STR=0x4000_1314, PF_STR=0x4000_1514, PG_STR=0x4000_1614

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		PST15	PST14	PST13	PST12	PST11	PST10	PST9	PST8	PST7	PST6	PST5	PST4	PST3	PST2	PST1	PST0
-		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	PSTx	Pin drive strength setting, x = 0–15	
		0	Disables the strength.
		1	Enables the strength.

NOTE: Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.7 Pn_IER: PORT n interrupt enable register

All pins can be set as an external interrupt source, either the edge-triggered interrupt or the level-triggered interrupt. The Pn_IER register enables or disables these interrupts for each pin.

PA_IER=0x4000_1020, PB_IER=0x4000_1120, PC_IER=0x4000_1220
PD_IER=0x4000_1320, PF_IER=0x4000_1520, PG_IER=0x4000_1620

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	The interrupt to be enabled for the pin, x = 0–15	
		00	Enables no interrupts.
		01	Enables the level-triggered interrupt (non-pending).
		10	Enables the level-triggered interrupt (pending).
		11	Enables the edge-triggered interrupt.

NOTE: Values of the Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can be changed without activating PORTEN.

In non-pending mode, you can check if the current state of the pin is an interrupt state. In high level non-pending interrupt mode, an interrupt signal is generated when the pin state is high. The interrupt status flag can be checked with the ISR register. In high level non-pending mode, the interrupt status is automatically cleared. Therefore, when the pin state changes from high to low, the interrupt state is automatically cleared.

In the pending mode, you can check whether an interrupt has occurred, not the current state of the pin. In high level pending interrupt mode, an interrupt signal is generated when the pin state is high. The interrupt status flag can be checked with the ISR register. In high level pending mode, the interrupt status is not automatically cleared. Therefore, when the state of the pin is changed from high to low, the interrupt will continue to occur unless the user clears the interrupt flag. In the pending mode, once an interrupt occurs, the user must clear the interrupt flag directly because the state is maintained until the user clears the interrupt flag directly.

5.3.8 Pn_ISR: PORT n interrupt status register

When an interrupt is received by the MCU, the status of the interrupt can be checked in Pn_ISR. The Pn_ISR register informs of which interrupt event has occurred at each interrupt source pin. To clear an interrupt occurrence flag, write a 1 to the bit.

PA_ISR=0x4000_1024, PB_ISR=0x4000_1124, PC_ISR=0x4000_1224
PD_ISR=0x4000_1324, PF_ISR=0x4000_1524, PG_ISR=0x4000_1624

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
RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC

2x+1 2x	PISx	Pin interrupt status, x = 0–15	
		00	No interrupt event has occurred.
		01	The low-level interrupt or falling-edge interrupt event has occurred.
		10	The high-level interrupt or rising-edge interrupt event has occurred.
		11	Both the rising- and falling-edge interrupt events have been detected in edge-triggered interrupt mode. Level-triggered interrupt mode is not supported.

NOTES:

1. In rising and falling edge interrupt mode, the pin state is '10' when the rising edge occurs and the pin state is '01' when the falling edge occurs.
2. Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.9 Pn_ICR: PORT n interrupt control register

The Pn_ICR register controls each pin’s interrupt modes that define interrupt-triggering signals.

PA_ICR=0x4000_1028, PB_ICR=0x4000_1128, PC_ICR=0x4000_1228
PD_ICR=0x4000_1328, PF_ICR=0x4000_1528, PG_ICR=0x4000_1628

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	Pin interrupt mode, x = 0–15	
		00	Enables no external interrupts.
		01	Enables low-level or falling-edge interrupt mode.
		10	Enables high-level or rising-edge interrupt mode.
		11	Enables both rising- and falling-edge interrupt modes. Level-triggered mode is not supported.

NOTE: Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.10 Pn_ODR: PORT n output data register

The Pn_ODR register defines the output data from each pin when the port is in GPIO mode. This register’s settings represent each pin’s output level.

PA_ODR=0x4000_1030, PB_ODR=0x4000_1130, PC_ODR=0x4000_1230
PD_ODR=0x4000_1330, PF_ODR=0x4000_1530, PG_ODR=0x4000_1630

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	POD
-	0x0000
-	RW

15 0	POD	The pin’s output data configuration	
		0	Outputs a 0 if the port pin is in output mode.
		1	Outputs a 1 if the port pin is in output mode.

NOTE: Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.11 Pn_IDR: PORT n input data register

The Pn_IDR register allows each pin level of the port to be read. This register’s settings represent the input at each pin in logic input mode even though the pin is currently in another mode, except for in analog mode. If a pin is set to serve a special function, such as an ADC or clock sender, the register bit always reads 1. If a level interrupt has been enabled for a port’s pin through Pn_ICR, the pin’s input level can be checked by reading the Pn_IDR settings.

PA_IDR=0x4000_1034, PB_IDR=0x4000_1134, PC_IDR=0x4000_1234
PD_IDR=0x4000_1334, PF_IDR=0x4000_1534, PG_IDR=0x4000_1634

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		PID	
-		0x0000	
-		RO	

15 0	PID	The pin’s input data (represents the port’s current status)	
		0	The current pin input is 0.
		1	The current pin input is 1.

NOTES:

1. If an external pin of the MCU is used as an input pin while it is floating, it will fall in an “unknown” state, where the input signal level is identified as either 0 or 1, regardless of the actual level.
2. If the MCU is programmed to perform a certain function based on the level identified by Pn_IDR in this floating state, this is highly likely to cause malfunctions. Therefore, it is recommended to pull up or down the pins that receive external input signals, depending on their uses.
3. Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.12 Pn_BSR: PORT n set/reset register

The Pn_BSR register sets or clears the level status flag of each pin belonging to Port n. The values written to Pn_BSR are read by Pn_ODR.

PA_BSR=0x4000_1038, PB_BSR=0x4000_1138, PC_BSR=0x4000_1238
PD_BSR=0x4000_1338, PF_BSR=0x4000_1538, PG_BSR=0x4000_1638

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
BCD15	BCD14	BCD13	BCD12	BCD11	BCD10	BCD9	BCD8	BCD7	BCD6	BCD5	BCD4	BCD3	BCD2	BCD1	BCD0	BSD15	BSD14	BSD13	BSD12	BSD11	BSD10	BSD9	BSD8	BSD7	BSD6	BSD5	BSD4	BSD3	BSD2	BSD1	BSD0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

X+16	BCDx	Pin clear, x = 16–31	
		0	Causes no changes.
		1	Resets the corresponding pin bit.
X	BSDx	Pin set, x = 0–15	
		0	Causes no changes.
		1	Sets the corresponding pin bit.

NOTE: Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.13 Pn_BCR: PORT n reset register

The Pn_BCR register clears the level status flag of each pin belonging to Port n. Setting a bit to 1 clears the corresponding bit in Pn_ODR.

PA_BCR=0x4000_103C, PB_BCR=0x4000_113C, PC_BCR=0x4000_123C
PD_BCR=0x4000_133C, PF_BCR=0x4000_153C, PG_BCR=0x4000_163C

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																BCD15	BCD14	BCD13	BCD12	BCD11	BCD10	BCD9	BCD8	BCD7	BCD6	BCD5	BCD4	BCD3	BCD2	BCD1	BCD0
-																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	BCDx	Pin clear, x = 0–15	
		0	Causes no changes.
		1	Resets the corresponding pin bit.

NOTE: Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.

5.3.14 PORTEN: PORT access enable register

The PORTEN register determines whether or not to allow changes to the settings of all PCU registers.

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																							ENS	PORTEN							
-																							0	--							
-																							RO	WO							

8	ENS	Whether the register is enabled or disabled
		0 Disables PORTEN.
		1 Enables PORTEN (PCU registers can be accessed).
7	PORTEN	Writing 0x15 and then 0x51 to the bit field enables writing new values to PCU registers. After this, write a different value to this bit field to protect the PCU registers against being updated with new values.

NOTES:

1. Pn_MR1, Pn_MR2, Pn_CR, and Pn_PRCR are registers that can change the value when PORTEN is activated.
2. Pn_DER, Pn_STR, Pn_IER, Pn_ISR, Pn_ICR, Pn_ODR, Pn_IDR, Pn_BSR and Pn_BCR registers can change the value of the register without activating PORTEN.
3. Example code using PORTEN

```
PORTEN=0x15; PORTEN=0x51; // Enables PORTEN.
                          // Pn_MR1,2, Pn_CR Pn_PRCR become settable
PORTEN=0                  // Disables PORTEN.
```

5.4 Functional description

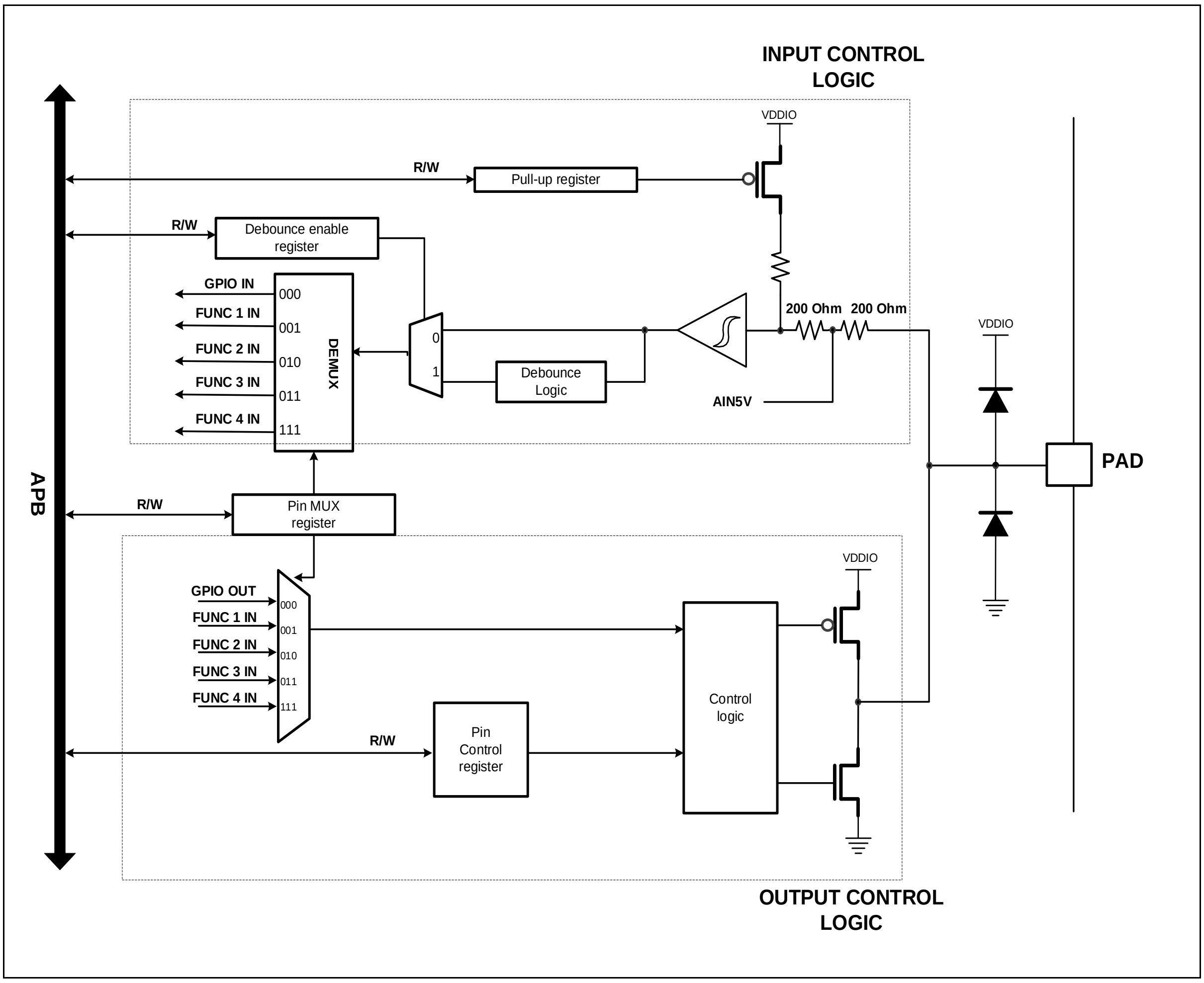
5.4.1 Port functionality

For the A33M11x series, all pins excluding certain specific-purpose pins can be used as GPIO pins. This GPIO function can be set in each port's pin mux register.

If an I/O port's pin is set as an input pin in the pin control register, its output function becomes disabled.

You can select different functions for each port' pins based on the settings of the alternative function selection register. The input data registers capture data inputted to each I/O pin, as either undebounced or debounced, at every GPIO clock cycle.

Figure 31. Port Function Block Diagram



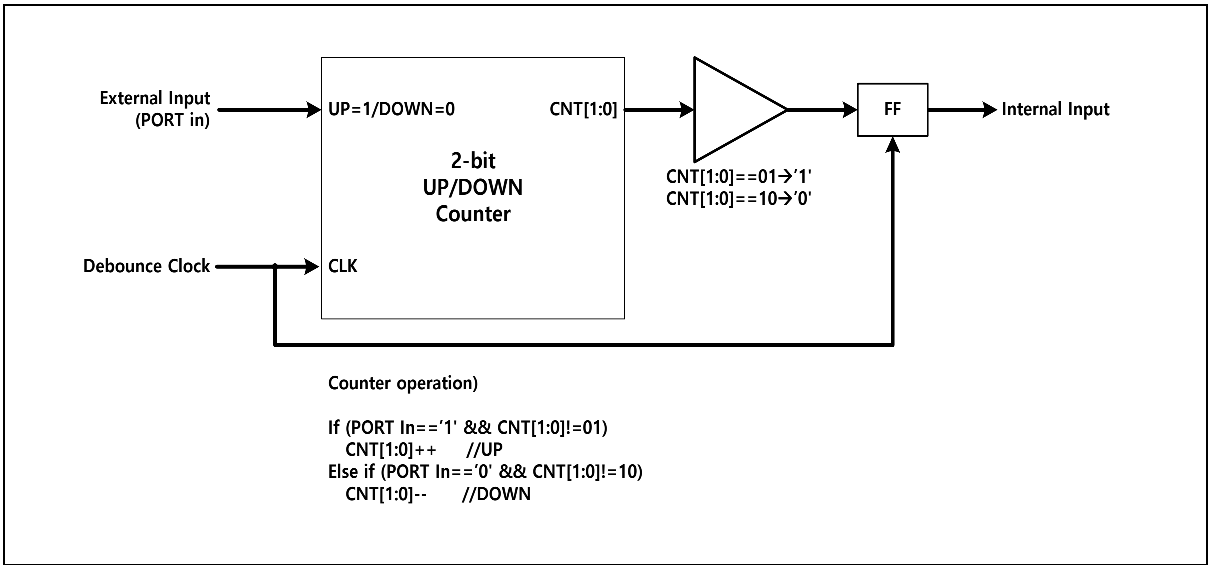
5.4.2 Debouncing functionality

For the A33M11x series, each port is capable of debouncing input signals. Debouncing is to filter out noises that can interfere with the port’s data input. Filtering levels can be adjusted for each 16-pin port, and each individual pin can be configured regarding whether to enable or disable filtering.

The debouncing clock used for each port can be set in SCU_MCCR4 and SCU_MCCR5.

The figure below shows a simplified block configuration.

Figure 32. Debouncing Logic



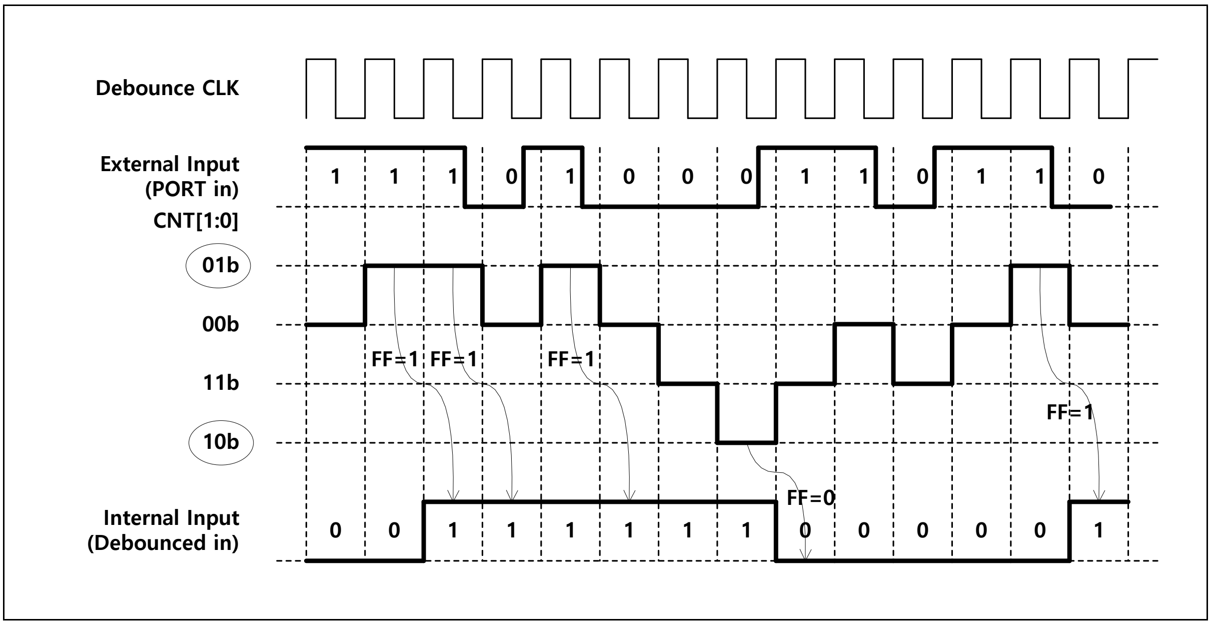
```
Counter operation)

If (PORT In=='1' && CNT[1:0]!=01)
   CNT[1:0]++     //UP
Else if (PORT In=='0' && CNT[1:0]!=10)
   CNT[1:0]--     //DOWN
```

When enabled, the debouncing logic is clocked at the frequency division ratio set in SCU_MCCR4 or SCU_MCCR5, depending on which register the port belongs to.

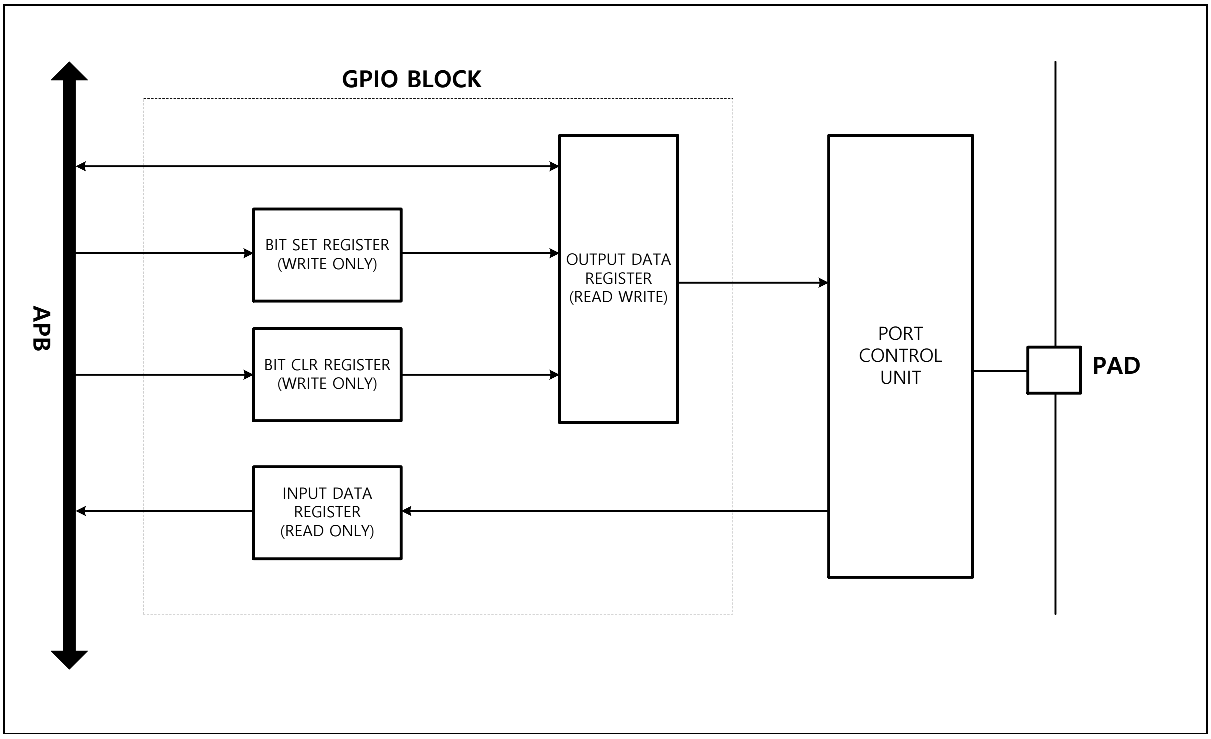
A change in input to a pin is recognized once the changed value is maintained constant for three clock pulses. If the internal clock is set to 01b, the internal input is set to 1; if the former is set to 10b, the latter is set to 0. Otherwise, the previous input value is maintained.

Figure 33. Port Debouncing Example



5.4.3 I/O block functionality

Figure 34. Port Function Block Diagram



GPIO pins in output mode output signals whose values are defined by the output data register (Pn_ODR). Once a pin’s output signal level is set in the pin bit set register (Pn_BSR), the corresponding bit in Pn_ODR becomes high. And once a pin’s output signal level is reset in the pin bit clear register (Pn_BCR), the corresponding bit in Pn_ODR becomes low.

The input data registers (Pn_IDRs) capture data inputted to each I/O pin, as either undebounced or debounced, at every GPIO clock cycle.

5.4.4 Interrupt functionality

The GPIO block has six interrupt sources, and each port can be an interrupt source. To set each port's GPIO pins as interrupt sources, you must set the polarity of edge or level that triggers an interrupt and configure the interrupt enable register (Pn_IER). A level-triggered or edge-triggered interrupt can be set for each pin. Once an interrupt occurs, the pin's corresponding Pn_ISR bit is flagged. This flag can be cleared by writing a 1 to the bit.

Table 23. GPIO Block's Interrupt Sources and Corresponding Pins

Interrupt name	Configurable pins
PA	Port A[15:0]
PB	Port B[15:0]
PC	Port C[15:0]
PD	Port D[3:0]
PF	Port F[4:0],[7:6]
PG	Port G[10]

5.4.5 Setting examples

<Example 1> PA0 input pin configuration for the falling-edge interrupt

```
PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                     : Enables PORTEN (enter 0x15 and then 0x51).

PA_MR1<P0MUX[1:0]> = “00”                           : Sets PA0 as a GPIO pin.
PA_CR<P0[1:0]> = “10”                               : Sets PA0 as an input pin.
PA_PRCR<PUE0[1:0]>= “10”                            : Enables pull-up for PA0.
PA_ICR<PIC0[1:0]>= “01”                             : Selects falling-edge interrupt mode.
PA_IER<PIE0[0:1]> = “11”                            : Enables the edge-triggered interrupt.
```

<Example 2> Debouncing enablement for PA0 pin

```
PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                     : Enables PORTEN (enter 0x15 and then 0x51).

SCU_MCCR4<PGADCSEL[26:24]> = “110”                  : Selects the clock source for PA port debouncing (HSE: 8MHz).
SCU_MCCR4<PGADCDIV[23:16]> = “1”                    : Sets the PA port’s debouncing clock divider (125 ns @ 8MHz).

PA_DER<PDE0[0]>= “1”                                : Enables debouncing for PA0
```

<Example 3> Configuration of pin function, I/O dir., and Pull-Up/Down resistor for the PD0 pin

```
PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                     : Enables PORTEN (enter 0x15 and then 0x51).

PD_MR1<P0MUX[1:0]> = “00”                           : Sets PD0 as a GPIO pin.
PD_CR<P0[1:0]> = “00”                               : Sets PD0 as an output pin.

PD_PRCR<PUE0[1:0]>= “11”                            : Enables pull-down for PD0.
```

<Example 4> Output data configuration for the PD0 pin

```
PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                     : Enables PORTEN (enter 0x15 and then 0x51).

PD_MR1<P0MUX[1:0]> = “00”                           : Sets PD0 as a GPIO pin.
PD_CR<P0[1:0]> = “00”                               : Sets PD0 as an output pin.

PD_ODR<POD[15:0]>= “00000000_00000001”              : Outputs the high-level signal from PD0  (This setting is valid
                                                    only with the pin set as an output pin).
```

6 Flash Memory Controller (FMC)

The Flash Memory Controller (FMC) is an interface controller of internal Flash memories:

- Flash code memory with 128-KB, and 256-KB protection bits
- 512B, and 2KB erases
- 128KB, and 256KB bulk erases
- Max available clock frequency : 28MHz
- Zero wait (less than 28MHz), 1- to 15-wait, and cache (Flash acceleration) access

Table 24. Code Flash Memory Controller Features

Item	Description	
Size	128KB	256KB
Start address	0x0000_0000	0x0000_0000
End address	0x0002_0000	0x0004_0000
Page size	512-byte	512-byte
Total page count	256 pages	512 pages
PGM unit	4-byte (1 word)	4-byte (1 word)
Erase unit	512-byte/ 2KB/ bulk	512-byte/ 2KB/ bulk

Figure 35. Code Flash Memory Map: 256KB Code Flash

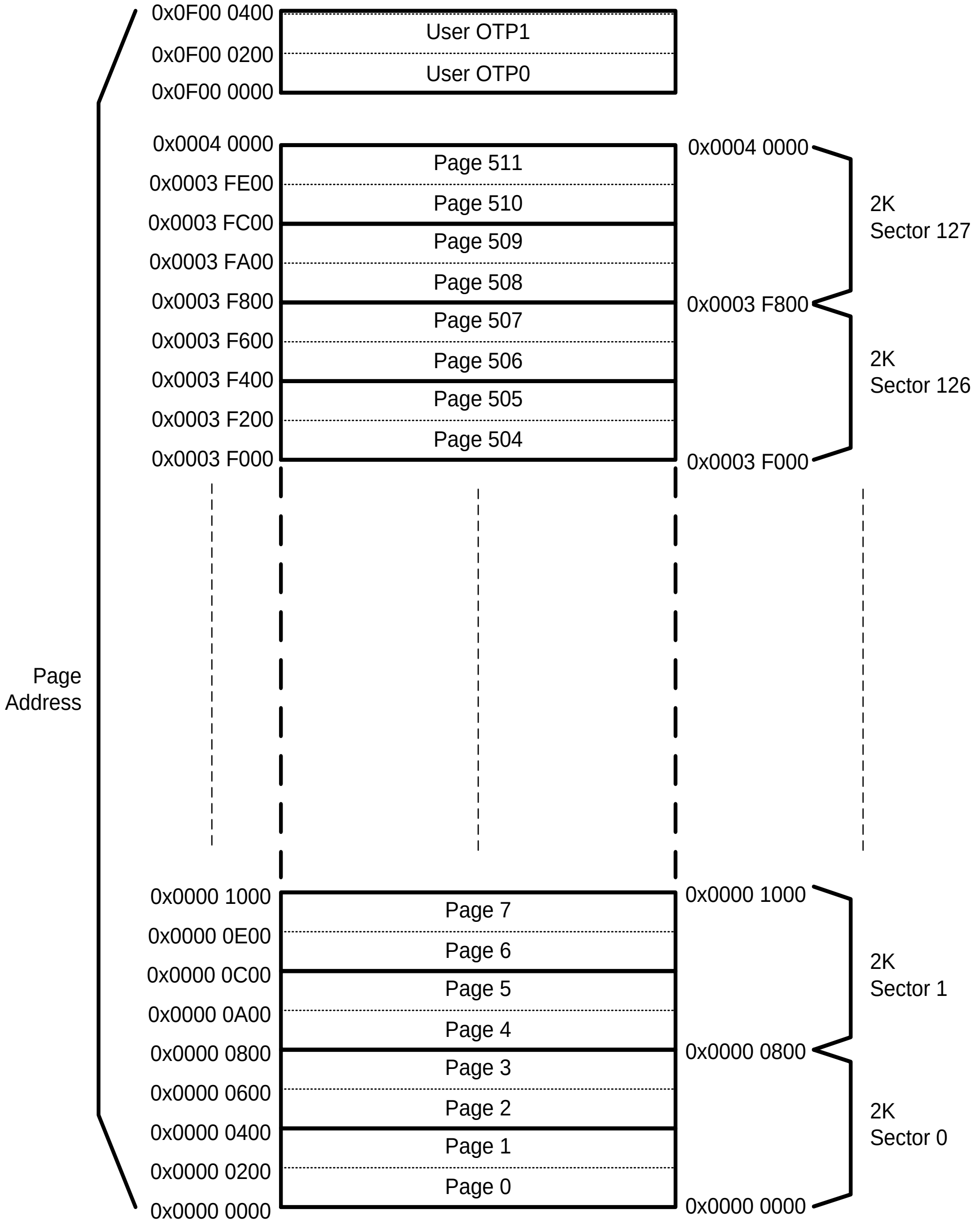
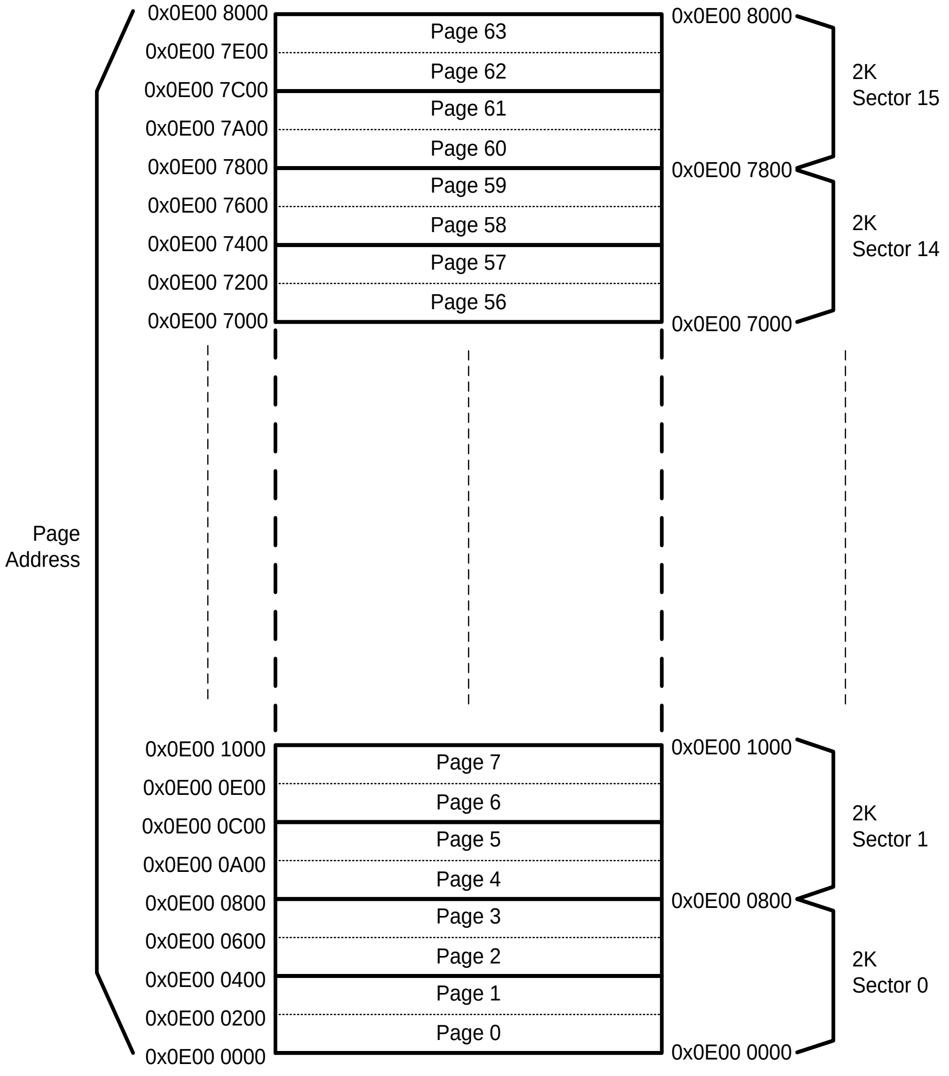


Table 25. Data Flash Memory Controller Features

Item	Description
Size	32KB
Start Address	0x0E00_0000
End Address	0x0E00_8000
Page Size	512-byte
Total Page Count	64 pages
PGM Unit	4-byte (1 word)
Erase Unit	512-byte / 2KB / bulk

Figure 36. Data Flash Memory Map: 32KB Data Flash



6.1 Code Flash registers

Tables shown below introduce the default address of the code Flash memory controller (CFMC):

Table 26. Base Address of Code Flash Memory Controller

Name	Base address
CFMC	0x4100_0000

Table 27. CFMC Register Map

Name	Offset	Type	Description	Reset value	Ref.
CFMC_CONF	0x0000	RW	Code Flash control register	0x0000_0000	6.1.1
CFMC_FLSKEY	0x0004	WO	Code Flash access key register	0x0000_0000	6.1.2
CFMC_OTPKEY	0x0008	WO	Code Flash OTP access key register	0x0000_0000	6.1.3
CFMC_FLSPROT	0x000C	RW	Code Flash protection register	0x0000_0000	6.1.4
CFMC_OTPPROT	0x0010	RW	Code Flash OTP protection register	0x0000_0000	6.1.5
CFMC_CTRL	0x0014	RW	Code Flash access control register	0xC000_0000	6.1.6
CFMC_STAT	0x0018	RW	Code Flash access status register	0x0000_0000	6.1.7
CFMC_READPROT	0x001C	RW	Code Flash read protection register	0x0000_00FF	6.1.8
CFMC_PWIN	0x0020	WO	Code Flash password input register	0x0000_0000	6.1.9
CFMC_CHKCTRL	0x0030	RW	Code Flash checksum control register	0x0000_0000	6.1.10
CFMC_CHKDOUT	0x0034	RW	Code Flash checksum data output register	0x0000_FFFF	6.1.11
CFMC_CHKSADDR	0x0038	RW	Code Flash checksum start address register	0x0000_0000	6.1.12
CFMC_CHKEADDR	0x003C	RW	Code Flash checksum end address register	0x0003_FFFF	6.1.13
CFMC_WTIMEOUT	0x003C	RW	Code Flash write timeout register	0x0000_0000	6.1.14
CFMC_PWPRST	0x0F18	WO	Code Flash password preset register	0x0000_0000	6.1.15

6.1.1 CFMC_CONF: code Flash control register

CFMC_CONF is the internal Flash memory’s control register. It is a 32-bit register.

CFMC_CONF=0x4100_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	WRITE	Reserved	DCRST	ICRST	Reserved	DCEN	ICEN	Reserved	LATENCY
-	0	-	0	0	-	0	0	-	0000
-	RW	-	RW	RW	-	RW	RW	-	RW

25	WRITE	Code Flash write enable bit	
		0	Disables.
		1	Enables.

NOTE: If this bit is not set to 1, the register value cannot be written to the code Flash register.

17	DCRST	Code Data cache reset control bit	
		0	No effect
		1	Reset
16	ICRST	Instruction cache reset control bit	
		0	No effect
		1	Reset
9	DCEN	Code Data cache enable bit	
		0	Disables.
		1	Enables.
8	ICEN	Instruction cache enable bit	
		0	Disables.
		1	Enables.
3 0	LATENCY	Flash wait value	
		The available wait values range from 0 to 15.	

NOTES: Cache setting procedure:

1. Disable instruction/ code data cache.
2. Reset instruction/ code data cache.(First, reset by setting the DCRST and ICRST bit to '1', and then clear the reset state by setting the DCRST and ICRST bit to '0'.)
3. Enable instruction/ code data cache.

Table 28. Internal Flash Access Time by Operating Clock

Wait Setting	Flash Access Wait	Max. Available Clock Frequency
0000	0-clock wait	Up to 28MHz
0001	1-clock wait	Up to 56MHz
0010	2-clock wait	Up to 84MHz
0011	3-clock wait	Up to 96MHz
0100	4-clock wait	Up to 96MHz
0101	5-clock wait	Up to 96MHz
0110	6-clock wait	Up to 96MHz
0111	7-clock wait	Up to 96MHz
1000	8-clock wait	Up to 96MHz
1001	9-clock wait	Up to 96MHz
1010	10-clock wait	Up to 96MHz
1011	11-clock wait	Up to 96MHz
1100	12-clock wait	Up to 96MHz
1101	13-clock wait	Up to 96MHz
1110	14-clock wait	Up to 96MHz
1111	15-clock wait	Up to 96MHz

6.1.2 CFMC_FLSKEY: code Flash access key register

CFMC_FLSKEY is the internal Flash memory’s access key register. It is a 32-bit register

CFMC_FLSKEY=0x4100_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
FKEY			
0x0000			
WO			

31 0	FKEY	The bits to which key values are written for accessing the Flash memory. Values must be written to the register in the following order to access the Flash memory. (KEY1 → KEY2 → KEY3)	
		KEY1	0x01234567
		KEY2	0x12345678
		KEY3	0x23456789

6.1.3 CFMC_OTPKEY: code Flash OTP access key register

FMC_OTPKEY is an OTP access key register. It is a 32-bit register.

CFMC_OTPKEY=0x4100_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
OKEY			
0x0000			
WO			

31 0	OKEY	The bits to which key values are written for accessing the OTP area. Values must be written to the register in the following order to access the Flash memory. (KEY1 → KEY2 → KEY3)	
		KEY1	0x3456789A
		KEY2	0x456789AB
		KEY3	0x56789ABC

6.1.4 CFMC_FLSPROT: code Flash protection register

CFMC_FLSPROT is an internal memory protection register. This register is 32 bits wide.

CFMC_FLSPROT=0x4100_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
FUP512B_7	FUP512B_6	FUP512B_5	FUP512B_4	FUP512B_3	FUP512B_2	FUP512B_1	FUP512B_0	Reserved								FPBY16K_15	FPBY16K_14	FPBY16K_13	FPBY16K_12	FPBY16K_11	FPBY16K_10	FPBY16K_9	FPBY16K_8	FPBY16K_7	FPBY16K_6	FPBY16K_5	FPBY16K_4	FPBY16K_3	FPBY16K_2	FPBY16K_1	FPBY16K_0
1	1	1	1	1	1	1	1	-								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	RW	RW	RW	RW	RW

X+24	FUP512B_x (x=0–7)	Unprotection of the last 4 KB of Flash memory (The settings of these bits have a higher priority level than the protection selection bits)	

		X=0: 0x0003_F000 – 0x0003_F1FF
		X=2: 0x0003_F400 – 0x0003_F5FF
		X=4: 0x0003_F800 – 0x0003_F9FF
		X=6: 0x0003_FC00 – 0x0003_FDFF
			X=1: 0x0003_F200 – 0x0003_F3FF
			X=3: 0x0003_F600 – 0x0003_F7FF
			X=5: 0x0003_FA00 – 0x0003_FBFF
			X=7: 0x0003_FE00 – 0x0003_FFFF

		0	Disable
		1	Enable
n	FPBY16K_n (n=0–15)	Flash area protection selection bits	

		N=0: 0x0000_0000 – 0x0000_3FFF
		N=2: 0x0000_8000 – 0x0000_BFFF
		N=4: 0x0001_0000 – 0x0001_3FFF
		N=6: 0x0001_8000 – 0x0001_BFFF
		N=8: 0x0002_0000 – 0x0002_3FFF
		N=10: 0x0002_8000 – 0x0002_BFFF
		N=12: 0x0003_0000 – 0x0003_3FFF
		N=14: 0x0003_8000 – 0x0003_BFFF
			N=1: 0x0000_4000 – 0x0000_7FFF
			N=3: 0x0000_C000 – 0x0000_FFFF
			N=5: 0x0001_4000 – 0x0001_7FFF
			N=7: 0x0001_C000 – 0x0001_FFFF
			N=9: 0x0002_4000 – 0x0002_7FFF
			N=11:0x0002_C000 – 0x0002_FFFF
			N=13: 0x0003_4000 – 0x0003_7FFF
			N=15: 0x0003_C000– 0x0003_FFFF

		0	Unprotection
		1	Protection

NOTES:

1. When a protection bit is enabled, chip erase does not work because the protected area is included in the erase.
2. Likewise, as each unprotection sector is 512 bytes, other erase commands (2K, and chip erases) do not work.
3. '0' (Disable) of FUP512B_x area is not Protection. It is only meaningful to unprotection of FUP512B_x area with '1' (Enable) in Protection status.

6.1.5 CFMC_OTPPROT: code Flash OTP protection register

CFMC_OTPPROT is an OTP protection register. It is a 32-bit register.

CFMC_OTPPROT=0x4100_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																														OP1	OP0
-																														0	0
-																														RW	RW

n	OPn
	(x=0~1)
		OTP area protection
		x=0: 0x0F00_0000 – 0x0F00_01FF
		x=1: 0x0F00_0200 – 0x0F00_03FF	

		0	Unprotection
		1	Protection

6.1.6 CFMC_CTRL: code Flash access control register

CFMC_CTRL is the internal Flash memory’s access control register. It is a 32-bit register.

CFMC_CTRL=0x4100_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
FLOCK	OLOCK	Reserved													BURSTMD	Reserved							WDIEN	WABOART	Reserved			CERS	SERS	PERS	PGM
0	0	-													0	-							0	0	-			0	0	0	0
RO	RO	-													RW	-							RW	RW	-			RW	RW	RW	RW

31	FLOCK	Flash lock	
		0	Unlock status
		1	Lock status
30	OLOCK	OTP lock	
		0	Unlock status
		1	Lock status
16	BURSTMD	Burst Flash write mode	
		0	Disables
		1	Enables
8	WDIEN	Whether to enable or disable the write done interrupt	
		0	Disables.
		1	Enables.
7	WABOART	Flash memory write abort mode	
		0	Disables.
		1	Enables.
3	CERS	Flash memory chip erase mode	
		0	Disables.
		1	Enables.
2	SERS	Flash memory 2 KB-sector erase mode	
		0	Disables.
		1	Enables.
1	PERS	Flash memory page erase mode	
		0	Disables.
		1	Enables.
0	PGM	Flash memory write	
		0	Disables.
		1	Enables.

NOTE: To disable the Flash lock or the OTP lock, the correct key values must be entered in the correct order by using the CFMC_FLSKEY or CFMC_OTPKEY register, respectively.

6.1.7 CFMC_STAT: code Flash access status register

CFMC_STAT is the internal Flash memory’s access status register. It is a 32-bit register.

CFMC_STAT=0x4100_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									WTERR	RPERR	WSERR	OPERR	FPERR	OLERR	FLERR	Reserved						CDONE	WDONE	Reserved						CBUSY	WBUSY
-									0	0	0	0	0	0	0	-						0	0	-						0	0
-									RC	RC	RC	RC	RC	RC	RC	-						RC	RC	-						RO	RO

22	WTERR	Write timeout error	
		0	Not error
		1	Error detection (cleared by writing a 1)
21	RPERR	Read-protect error	
		0	Not error
		1	Error detection (cleared by writing a 1)
20	WSERR	Write sequence error	
		0	Not error
		1	Error detection (cleared by writing a 1)
19	OPERR	OTP protect error	
		0	Not error
		1	Error detection (cleared by writing a 1)
18	FPERR	Flash protect error	
		0	Not error
		1	Error detection (cleared by writing a 1)
17	OLERR	OTP lock error	
		0	Not error
		1	Error detection (cleared by writing a 1)
16	FLERR	Flash lock error	
		0	Not error
		1	Error detection (cleared by writing a 1)
9	CDONE	Checksum done check	
		0	Busy
		1	Checksum done (cleared by writing a 1)
8	WDONE	Write done interrupt status	
		0	Busy
		1	Write done (cleared by writing a 1)
1	CBUSY	Checksum busy check	
		0	Not busy
		1	Busy
0	WBUSY	Write busy check	
		0	Not busy
		1	Busy

6.1.8 CFMC_READPROT: code Flash read protection register

CFMC_READPROT is the internal Flash memory's read protection register. It is a 32-bit register. When read protection is enabled, read protection is applied to all areas of the code Flash, and Flash access and debug operations are restricted depending on the protection level.

If read protection is applied, it can be released in two ways as follows. In the first method, when the preset function is applied to the CFMC_PWPRST register together with the write protection function, the password input value set in the CFMC_PWIN register is input. When the input password matches, the read protection function is released. The other way is to erase the read protection by erasing the data of all code Flash areas with Chip Erase and OTP0 Erase.

CFMC_READPROT=0x4100_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
DBGMOD	SRBOOT	Reserved	PWMATCH	Reserved	CERSD	Reserved	LVL2_STS	LVL1_STS	Reserved	LVL2_EN	LVL1_EN	RPROT
0	0	-	0	-	0	-	0	0	-	0	0	0xFF
RO	RO	-	RO	-	RO	-	RO	RO	-	RO	RO	RW

31	DBGMOD	Debug Operating Status bit	
		0	Not operating
		1	Operating
30	SRBOOT	SRAM Boot Mode Status bit	
		0	Normal Mode
		1	SRAM Boot Mode
26	PWMATCH	Password Match Flag bit	
		0	Not Matched
		1	Password Preset value, Password In value Matched.
24	CERSD	Chip Erase Done Flag bit	
		0	Not Occurred
		1	Chip Erase Done (OTP0 Erase/Write Permit)
17	LVL2_STS	Protection Level 2 Status bit (raw data)	
		0	Normal Status
		1	Protection Level2 Status
16	LVL1_STS	Protection Level 1 Status bit (raw data)	
		0	Normal Status
		1	Protection Level1 Status
9	LVL2_EN	Protection Level 2 Enable Status Bit	
		0	Disable Status
		1	Enable Status
8	LVL1_EN	Protection Level 1 Enable Status Bit	
		0	Disable Status
		1	Enable Status
7 0	RPROT	Read Protection Control Bit	
		0xFF	Unprotection
		0x39	Protection Level 1
		others	Protection Level 2
		* When the memory is in either protection mode, setting the 7th bit to 1 enables the password function for the protection mode. Ex) 0xB9: Protection 1 Password Mode	

	0x80: Protection 2 Password Mode

Table 29. Available Operating Modes by Protection Level

Protection level	Operation mode	Code main		OTP		Data main	
		read	write	read	write	read	write
UNPROT	Normal mode	O	O	O	O	O	O
	Debug mode	O	O	O	O	O	O
	Boot mode	O	O	O	O	O	O
LVL1	Normal mode	O	O	O	O	O	O
	Debug mode	X	X	X	X	X	X
	Boot mode	X	X	X	X	X	X
LVL2	Normal mode	O	O	O	O	O	O
	Debug mode	This mode cannot be entered.					
	Boot mode	X	X	X	X	X	X

NOTES:

1. The priority levels of protection mode are as follows:
   A. Normal Protection > Password Protection
   B. Normal Protection : RPROT2 > RPROT1 > UNRPROT
   C. Password Protection : RPROT2 > RPROT1 > UNRPROT

   A higher protection level cannot transition to a lower level without the chip erase and OTP0 erase executed.
2. Password protection mode can only be changed to Password protection mode and Normal protection mode can be changed to Normal protection mode.

   (You cannot switch from password protection mode to normal protection mode, from normal protection mode to password protection mode.)

   Example: Procedure for transitioning to UNRPROT from RPROT1
   A. Chip erase
   B. READPROT[24].CERSD flag check
   C. OTP0 erase
   D. Disable Read Protection mode at reset
3. If you want to erase OTP area when the protection level is set up, you should run chip erase first.

6.1.9 CFMC_PWIN: code Flash password input register

CFMC_PWIN is used to input a password when a password preset value is set in the CFMC_PWRRST register. When this value is matched with the preset value first input to the CFMC_PWPRST register, the protected data can be read. If a value is input to this register without specifying the password preset value, the input value is ignored.

CFMC_PWIN=0x4100_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
PWIN																															
0x0000																															
WO																															

31 0	PWIN	Password input data bit
		Writing is done twice to the register in the order from LSB to MSB. Rewriting is restricted once the register is written.

NOTES:

1. How to specify the password value as 0x1234_5678, 0x8765_4321.

```
CFMC_PWIN = 0x1234_5678;
CFMC_PWIN = 0x8765_4321;
```

2. When Read Protection is released with Password input, two times input of the same Password will activate the Read Protection immediately.

```
CFMC_PWIN = 0x1234_5678;
CFMC_PWIN = 0x8765_4321; // Unprotection Read Protection Mode with
 Password Match
CFMC_PWIN = 0x1234_5678;
CFMC_PWIN = 0x8765_4321; // Reset to Read Protection Password mode
```

3. If you enter the wrong password value when you enter the first password, you must reset it after resetting.

```
CFMC_PWIN = 0x1111_1111; // Enter incorrect password value
CFMC_PWIN = 0x1234_5678;
CFMC_PWIN = 0x8765_4321; // Password is not released
```

6.1.10 CFMC_CHKCTRL: code Flash checksum control register

CFMC_CHKCTRL is the internal Flash memory's checksum control register. It is a 32-bit register. The checksum function is executable in read protection mode as well.

CFMC_CHKCTRL=0x4100_0030

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td>CDRST</td><td colspan="7">Reserved</td><td>CDIEN</td><td colspan="6">Reserved</td><td>BSTEN</td><td>BGEN</td></tr>
<tr><td colspan="15">-</td><td>0</td><td colspan="7">-</td><td>0</td><td colspan="6">-</td><td>0</td><td>0</td></tr>
<tr><td colspan="15">-</td><td>RW</td><td colspan="7">-</td><td>RW</td><td colspan="6">-</td><td>RW</td><td>RW</td></tr>
</table>

16	CDRST	Checksum data reset (automatically cleared)	
		0	No Effect
		1	Reset
8	CDIEN	Checksum done interrupt	
		0	Disables.
		1	Enables.
1	BSTEN	Burst mode	
		0	Disables.
		1	Enables.
0	BGEN	Background mode	
		0	Disables.
		1	Enables.

NOTE: Polynomial - 0x1021 (16-bit CCITT)

6.1.11 CFMC_CHKDOUT: code Flash checksum data output register

CFMC_CHKDOUT is the internal Flash memory's checksum data output register. It is a 32-bit register.

CFMC_CHKDOUT=0x4100_0034

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CDOUT</td></tr>
<tr><td colspan="16">-</td><td colspan="16">0xFFFF</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RO</td></tr>
</table>

15 0	CDOUT	Checksum data output

6.1.12 CFMC_CHKSADDR: code Flash checksum start address register

CFMC_CHKSADDR is the internal Flash memory’s checksum start address register. It is a 32-bit register.

CFMC_CHKSADDR=0x4100_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
SADDR																								FIXED VALUE							
0x000000																								0x00							
RW																															

31 8	SADDR	Checksum start address

6.1.13 CFMC_CHKEADDR: code Flash checksum end address register

CFMC_CHKEADDR is the internal Flash memory’s checksum end address register. It is a 32-bit register.

CFMC_CHKEADDR=0x4100_003C

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
EADDR																								FIXED VALUE							
0x0003FF																								0xFF							
RW																															

31 8	EADDR	Checksum end address

6.1.14 CFMC_WTIMEOUT: code Flash write timeout register

This register can set the Flash write time. If a write is not completed during "HCLK x WTIMEOUT" after the write (program, erase), the write will be aborted. This register is a 32-bit register.

CFMC_WTIMEOUT=0x4100_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
Reserved			WTIMEOUT
-			0x00
-			RW

7 0	WTIMEOUT	Flash write timeout configuration	
		0	Disables
		1	Enables

6.1.15 CFMC_PWPRST: code Flash password preset register

This register is used in combination with the Flash read protection function, and the PRESET value can be set when using the PASSWORD function. If a password is entered twice in this register, the password is set. The password specified during user code execution is initialized by a system reset. If you want to preserve the preset value even after system initialization, you must use the User Bootloader to assign a password to the preset in the User Data area in the system initialization step.

CFMC_PWPRST=0x4100_0F18

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
PWPRST			
0x0000			
WO			

31 0	PWPRST	Password preset data bit
		Writing is done twice to the register in the order from LSB to MSB. Rewriting is restricted once the register is written.

Example: Presetting 0x1234_5678 as a password

```
CFMC_PRST = 0x1234_5678;
CFMC_PRST = 0x1234_5678;
```

6.2 Data Flash registers

The table below shows the default address of the data Flash memory controller (DFMC).

Table 30. Base Address of Data Flash Memory Controller

Name	Base address
DFMC	0x4100_1000

Table 31. DFMC Register Map

Name	Offset	Type	Description	Reset value	Reference
DFMC_CONF	0x0000	RW	Data Flash control register	0x0000_0000	6.2.1
DFMC_FLSKEY	0x0004	WO	Data Flash access key register	0x0000_0000	6.2.2
DFMC_FLSPROT	0x000C	RW	Data Flash protection register	0x0000_0000	6.2.3
DFMC_CTRL	0x0014	RW	Data Flash access control register	0x8000_0000	6.2.4
DFMC_STAT	0x0018	RW	Data Flash access status register	0x0000_0000	6.2.5
DFMC_CHKCTRL	0x0030	RW	Data Flash checksum control register	0x0000_0000	6.2.6
DFMC_CHKDOUT	0x0034	RW	Data Flash checksum data output register	0x0000_FFFF	6.2.7
DFMC_CHKSADDR	0x0038	RW	Data Flash checksum start address register	0x0000_0000	6.2.8
DFMC_CHKEADDR	0x003C	RW	Data Flash checksum end address register	0x0000_7FFF	6.2.9
DFMC_WTIMEOUT	0x0040	RW	Data Flash write timeout register	0x0000_0000	6.2.10

6.2.1 DFMC_CONF: data Flash control register

DFMC_CONF is the internal Flash memory's control register. It is a 32-bit register.

DFMC_CONF=0x4100_1000

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	WRITE	Reserved	LATENCY
-	0	-	0000
-	RW	-	RW

25	WRITE	Data Flash write enable bit
		0 Disables.
		1 Enables.
NOTE: If this bit is not set to 1, the register value cannot be written to the code Flash register.		
3 0	LATENCY	Flash wait value
		The available wait values range from 0 to 15.

Table 32. Internal Flash Access Time by Operating Clock

Wait Setting	Flash Access Wait	Max. Available Clock Frequency
0000	0-clock wait	Up to 28MHz
0001	1-clock wait	Up to 56MHz
0010	2-clock wait	Up to 84MHz
0011	3-clock wait	Up to 96MHz
0100	4-clock wait	Up to 96MHz
0101	5-clock wait	Up to 96MHz
0110	6-clock wait	Up to 96MHz
0111	7-clock wait	Up to 96MHz
1000	8-clock wait	Up to 96MHz
1001	9-clock wait	Up to 96MHz
1010	10-clock wait	Up to 96MHz
1011	11-clock wait	Up to 96MHz
1100	12-clock wait	Up to 96MHz
1101	13-clock wait	Up to 96MHz
1110	14-clock wait	Up to 96MHz
1111	15-clock wait	Up to 96MHz

6.2.2 DFMC_FLSKEY: data Flash access key register

DFMC_FLSKEY is the internal Flash memory's access key register. It is a 32-bit register

DFMC_FLSKEY=0x4100_1004

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
FKEY			
0x0000			
WO			

31 0	FKEY	The bits to which key values are written for accessing the Flash memory. Values must be written to the register in the following order to access the Flash memory. (KEY1 → KEY2 → KEY3)	
		KEY1	0x01234567
		KEY2	0x12345678
		KEY3	0x23456789

6.2.3 DFMC_FLSPROT: data Flash protection register

DFMC_FLSPROT is an internal memory protection register. This register is 32 bits wide.

DFMC_FLSPROT=0x4100_100C

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
FUP512B_7	FUP512B_6	FUP512B_5	FUP512B_4	FUP512B_3	FUP512B_2	FUP512B_1	FUP512B_0	Reserved								FPBY2K_15	FPBY2K_14	FPBY2K_13	FPBY2K_12	FPBY2K_11	FPBY2K_10	FPBY2K_9	FPBY2K_8	FPBY2K_7	FPBY2K_6	FPBY2K_5	FPBY2K_4	FPBY2K_3	FPBY2K_2	FPBY2K_1	FPBY2K_0
0	0	0	0	0	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	RW	RW	RW	RW	RW

X+24	FUP512B_x
	(x=0–7)
		Unprotection of the first 4 KB of the Flash memory
		X=0: 0x0E00_7000 – 0x0E00_71FF
		X=1: 0x0E00_7200 – 0x0E00_73FF
		X=2: 0x0E00_7400 – 0x0E00_75FF
		X=3: 0x0E00_7600 – 0x0E00_77FF
		X=4: 0x0E00_7800 – 0x0E00_79FF
		X=5: 0x0E00_7A00 – 0x0E00_7BFF
		X=6: 0x0E00_7C00 – 0x0E00_7DFF
		X=7: 0x0E00_7E00 – 0x0E00_7FFF	

		0	Disable
		1	Enable

n	FPBY2K_n
	(n=0–15)
		Flash area protection
		N=0: 0x0E00_0000 – 0x0E00_07FF
		N=1: 0x0E00_0800 – 0x0E00_0FFF
		N=2: 0x0E00_1000 – 0x0E00_17FF
		N=3: 0x0E00_1800 – 0x0E00_1FFF
		N=4: 0x0E00_2000 – 0x0E00_27FF
		N=5: 0x0E00_2800 – 0x0E00_2FFF
		N=6: 0x0E00_3000 – 0x0E00_37FF
		N=7: 0x0E00_3800 – 0x0E00_3FFF
		N=8: 0x0E00_4000 – 0x0E00_47FF
		N=9: 0x0E00_4800 – 0x0E00_4FFF
		N=10: 0x0E00_5000 – 0x0E00_57FF
		N=11: 0x0E00_5800 – 0x0E00_5FFF
		N=12: 0x0E00_6000 – 0x0E00_67FF
		N=13: 0x0E00_6800 – 0x0E00_6FFF
		N=14: 0x0E00_7000 – 0x0E00_77FF
		N=15: 0x0E00_7800 – 0x0E00_7FFF	

		0	Unprotection
		1	Protection

6.2.4 DFMC_CTRL: data Flash access control register

DFMC_CTRL is the internal Flash memory’s access control register. It is a 32-bit register.

DFMC_CTRL=0x4100_1014

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
FLOCK	Reserved														BURSTMD	Reserved							WDIEN	WABOART	Reserved			CERS	SERS	PERS	PGM
0	-														0	-							0	0	-			0	0	0	0
RO	-														RW	-							RW	RW	-			RW	RW	RW	RW

31	FLOCK	Flash lock	
		0	Unlock status
		1	Lock status
16	BURSTMD	Burst Flash write mode	
		0	Disables
		1	Enables
8	WDIEN	Whether to enable or disable the write done interrupt	
		0	Disables.
		1	Enables.
7	WABOART	Flash memory write abort mode	
		0	Disables.
		1	Enables.
3	CERS	Flash memory chip erase mode	
		0	Disables.
		1	Enables.
2	SERS	Flash memory 2 KB-sector erase mode	
		0	Disables.
		1	Enables.
1	PERS	Flash memory page erase mode	
		0	Disables.
		1	Enables.
0	PGM	Flash memory write	
		0	Disables.
		1	Enables.

NOTE: To disable the Flash lock, the correct key values must be entered in the correct order by using the DFMC_FLSKEY register.

6.2.5 DFMC_STAT: data Flash access status register

DFMC_STAT is the internal Flash memory's access status register. It is a 32-bit register.

DFMC_STAT=0x4100_1018

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	WTERR	RPERR	WSERR	Reserved	FPERR	Reserved	FLERR	Reserved	CDONE	WDONE	Reserved	CBUSY	WBUSY
-	0	0	0	-	0	-	0	-	0	0	-	0	0
-	RC	RC	RC	-	RC	-	RC	-	RC	RC	-	RO	RO

22	WTERR	Write timeout error	
		0	Not error
		1	Error detection (cleared by writing a 1)
21	RPERR	Read-protect error	
		0	Not error
		1	Error detection (cleared by writing a 1)
20	WSERR	Write sequence error	
		0	Not error
		1	Error detection (cleared by writing a 1)
18	FPERR	Flash protect error	
		0	Not error
		1	Error detection (cleared by writing a 1)
16	FLERR	Flash lock error	
		0	Not error
		1	Error detection (cleared by writing a 1)
9	CDONE	Checksum done check	
		0	Busy
		1	Checksum done (cleared by writing a 1)
8	WDONE	Write done interrupt status	
		0	Busy
		1	Write done (cleared by writing a 1)
1	CBUSY	Checksum busy check	
		0	Not busy
		1	Busy
0	WBUSY	Write busy check	
		0	Not busy
		1	Busy

6.2.6 DFMC_CHKCTRL: data Flash checksum control register

DFMC_CHKCTRL is the internal Flash memory's checksum control register. It is a 32-bit register.

DFMC_CHKCTRL=0x4100_1030

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															CDRST	Reserved							CDIEN	Reserved						BSTEN	BGEN
-															0	-							0	-						0	0
-															RW	-							RW	-						RW	RW

16	CDRST	Checksum data reset (automatically cleared)	
		0	No Effect
		1	Reset
8	CDIEN	Checksum done interrupt	
		0	Disables.
		1	Enables.
1	BSTEN	Burst mode	
		0	Disables.
		1	Enables.
0	BGEN	Background mode	
		0	Disables.
		1	Enables.

NOTE: Polynomial - 0x1021 (16-bit CCITT)

6.2.7 DFMC_CHKDOUT: data Flash checksum data output register

DFMC_CHKDOUT is the internal Flash memory's checksum data output register. It is a 32-bit register.

DFMC_CHKDOUT=0x4100_1034

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																CDOUT															
-																0xFFFF															
-																RO															

15 0	CDOUT	Checksum data output

6.2.8 DFMC_CHKSADDR: data Flash checksum start address register

DFMC_CHKSADDR is the internal Flash memory’s checksum start address register. It is a 32-bit register.

DFMC_CHKSADDR=0x4100_1038

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">SADDR</td><td colspan="8">FIXED VALUE</td></tr>
<tr><td colspan="24">0x000000</td><td colspan="8">0x00</td></tr>
<tr><td colspan="24">RW</td><td colspan="8"></td></tr>
</table>

31 8	SADDR	Checksum start address

6.2.9 DFMC_CHKEADDR: data Flash checksum end address register

DFMC_CHKEADDR is the internal Flash memory’s checksum end address register. It is a 32-bit register.

DFMC_CHKEADDR=0x4100_103C

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="26">EADDR</td><td colspan="6">FIXED VALUE</td></tr>
<tr><td colspan="26">0x00001FF</td><td colspan="6">0x3F</td></tr>
<tr><td colspan="26">RW</td><td colspan="6"></td></tr>
</table>

31 6	EADDR	Checksum end address

6.2.10 DFMC_WTIMEOUT: data Flash write timeout register

This register can set the Flash write time. If a write is not completed during "HCLK x WTIMEOUT" after the write (program, erase), the write will be aborted. This register is a 32-bit register.

DFMC_WTIMEOUT=0x4100_1040

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																								WTIMEOUT							
-																								0x00							
-																								RW							

7 0	WTIMEOUT	Flash write timeout configuration	
		0	Disables
		1	Enables

6.3 Functional description

6.3.1 How to lock/unlock

1. To unlock an area in either Flash memory, you must enter key values in the key register in the following order:

   A. In normal condition: KEY1 → KEY2 → KEY3

   B. In abnormal condition: KEY1 → APB (no key) write → KEY2 → APB (no key) write → KEY3

2. A lock can be applied by writing to the corresponding control register.

   A. The CTRL register allows you to set your desired range of bits to be locked.

6.3.2 Sequence of resetting all caches

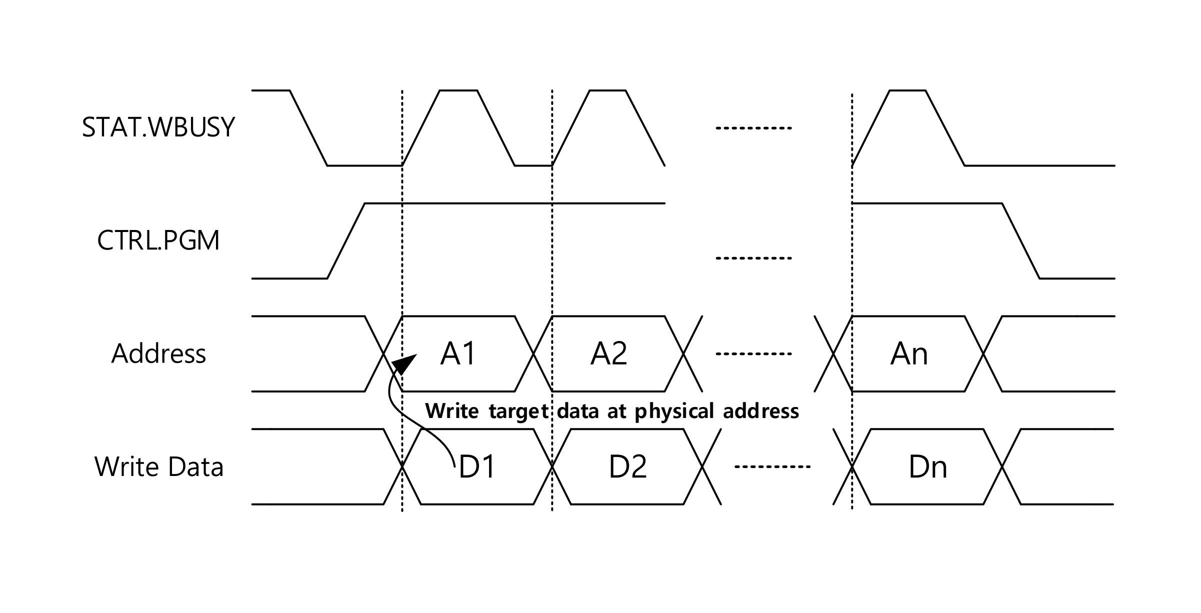
1. Disable instruction/code data cache
2. Reset instruction/code data cache
3. Enable instruction/code data cache

6.3.3 Writing sequence

Before writing to an area in either Flash memory, the area must be unlocked and unprotected. Once the writing is finished, make sure that the unlocked area is locked, the unprotected area is protected, and all (instruction/data) caches are reset.

6.3.3.1 Flash program example

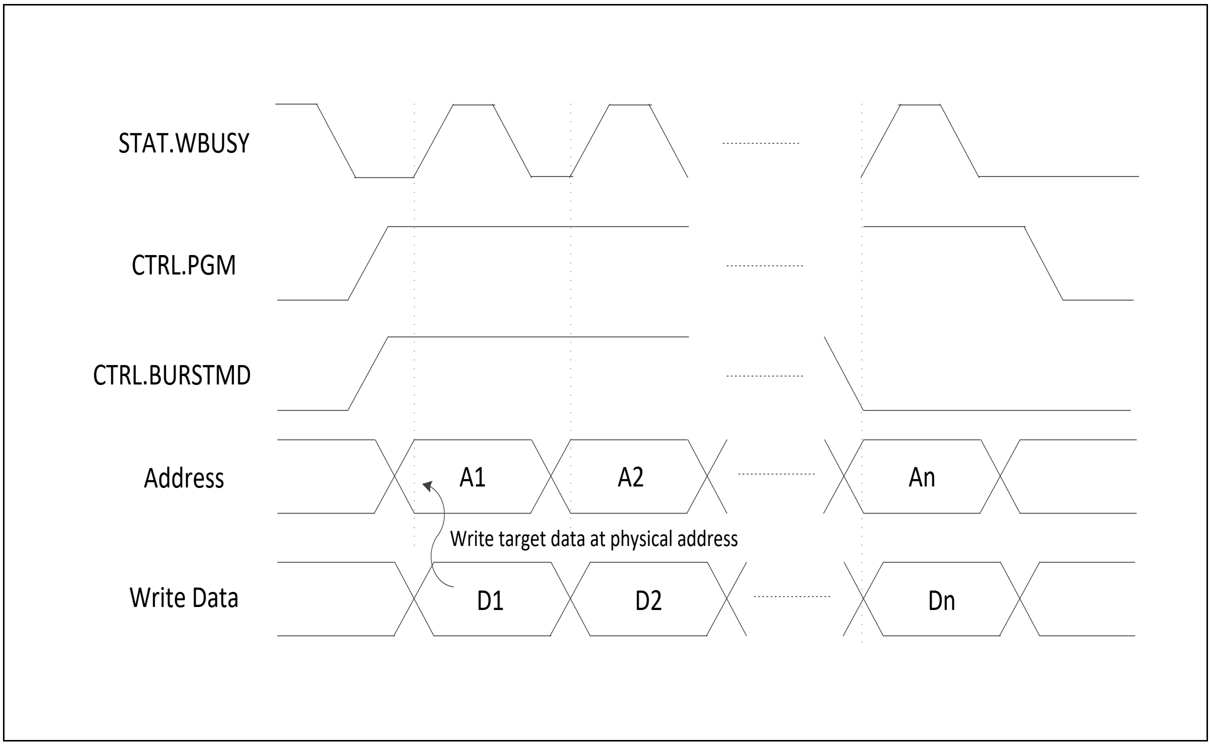
Figure 37. Flash Program Sequence



1. Check FMC_STAT.WBUSY
2. Set FMC_CTRL.PGM
3. Write target data at physical address
4. FMC_STAT.WBUSY polling or FMC_STAT WDONE interrupt
5. Clear FMC_CTRL.PGM

6.3.3.2 Flash burst program example

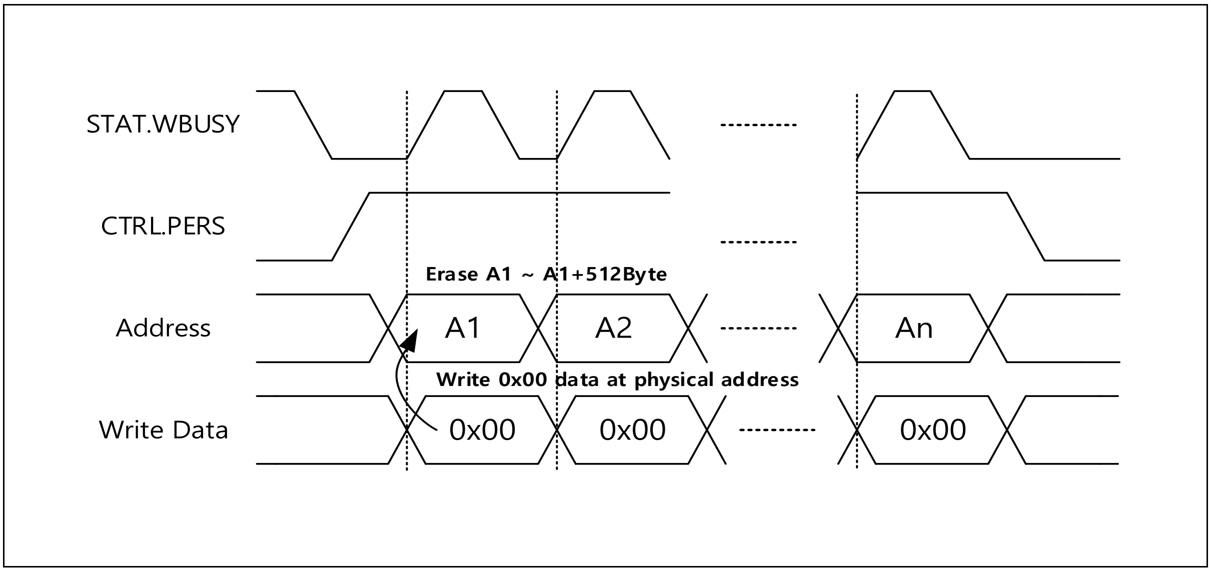
Figure 38. Flash Burst Program Sequence



1. Check FMC_STAT.WBUSY
2. Set FMC_CTRL.PGM and FMC_CTRL.BURSTMD
3. Write target data at physical address(Set FMC_CTRL.BURSTMD to 0 on the last data write)
4. FMC_STAT.WBUSY polling or FMC_STAT WDONE interrupt
5. Clear FMC_CTRL.PGM

6.3.3.3 Flash page erase example

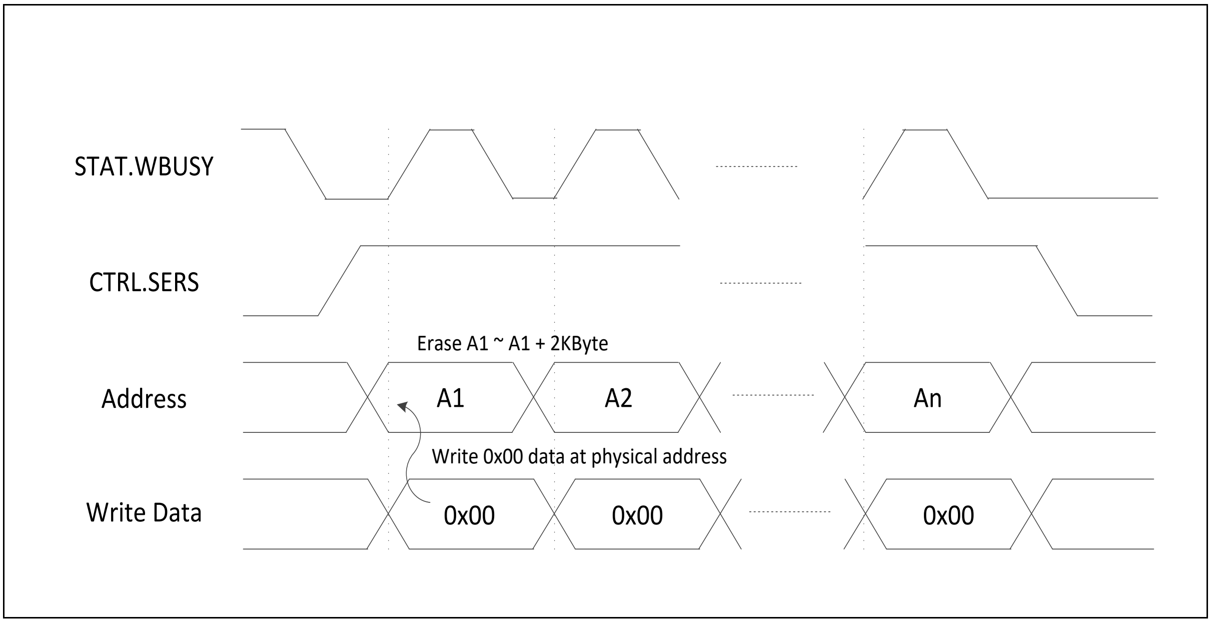
Figure 39. Flash Page Erase Sequence



1. Check FMC_STAT.WBUSY
2. Set FMC_CTRL.PERS
3. Write 0x0 data at physical page address
4. FMC_STAT.WBUSY polling or FMC_STAT.WDONE interrupt
5. Clear FMC_CTRL.PERS

6.3.3.4 Flash 2KB sector erase example

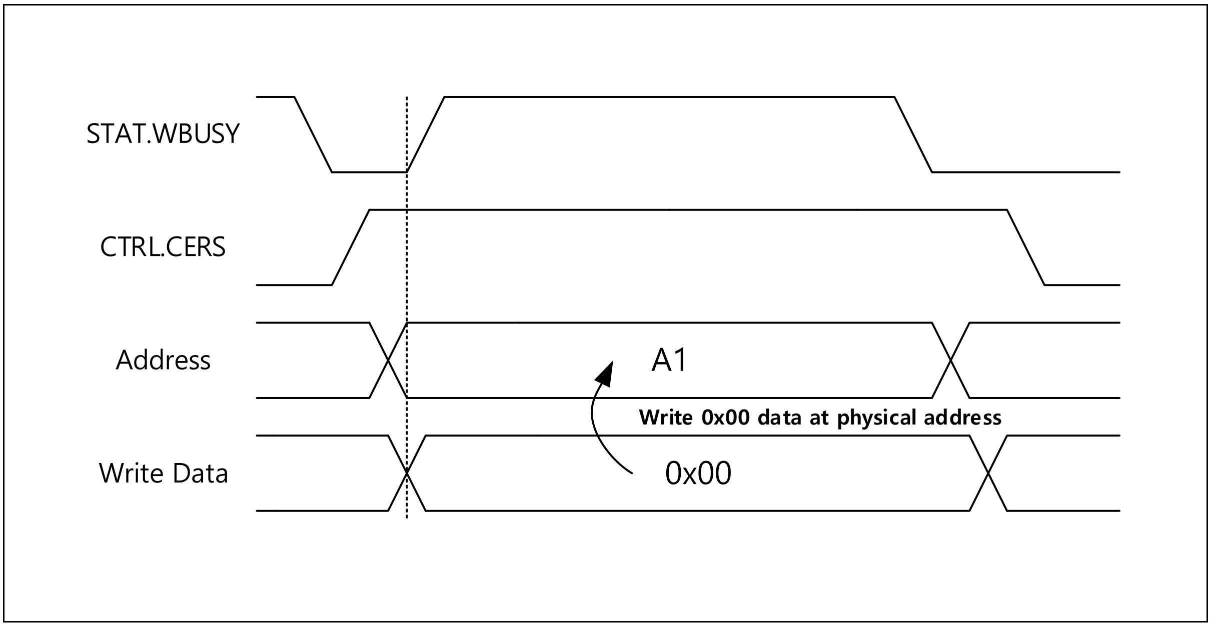
Figure 40. Flash 2KB Sector Erase Sequence



1. Check FMC_STAT.WBUSY
2. Set FMC_CTRL.SERS
3. Write 0x0 data at physical 2KB sector address
4. FMC_STAT.WBUSY polling or FMC_STAT.WDONE interrupt
5. Clear FMC_CTRL.SERS

6.3.3.5 Chip erase example

Figure 41. Flash Chip Erase Sequence



1. Check FMC_STAT.WBUSY
2. Set FMC_CTRL.CERS
3. Write 0x0 data at any physical address
4. FMC_STAT.WBUSY polling or FMC_STAT.WDONE interrupt
5. Clear FMC_CTRL.CERS

6.3.4 Checksum control sequence

16-bit ccitt data input bus size: 128 bits

1. Background mode (Checksum is running at only FLASH idle state)
    A. Set Checksum start address
    B. Set Checksum end address
    C. Reset Checksum data
    D. Enable BGEN, background
    E. FMC_STAT.CBUSY polling or FMC_STAT.CDONE interrupt
2. Burst mode (Checksum is running continuously with CPU halt state)
    A. Set Checksum start address
    B. Set Checksum end address
    C. Reset Checksum data
    D. Enable BSEN, Burst mode
    E. FMC_STAT.CBUSY polling or FMC_STAT.CDONE interrupt

6.3.5 Setting examples

<Example 1> Flash write

```
while (FMC_STAT<WBUSY[0]> == 1);                   : Checks that FMC is not busy.

FMC_CTRL<PGM[0]> = 1;                              : Enables programming mode.

For (i=0; i<0x1000; i=i+4){                        : Sets the address of the sector for write.
  (*(volatile unsigned int *) (i)) = 0xABCD;       : Inserts 0xABCD in address i.
  while (FMC_STAT<WBUSH[0]> == 1);}                : Checks that FMC is not busy.

FMC_CTRL<PGM[0]> = 0;                              : Disables programming mode.
```

<Example 2> 2Kbyte sector erase

```
While (FMC_STAT<WBUSY[0]> == 1);                   : Checks that FMC is not busy.

FMC_CTRL<SERS[2]> = 1;                             : Enables 2 KB-sector erase mode.

For (i=0; i<0x1000; i=i+400){                      : Sets the address of the sector for 2 KB erase.
  (*(volatile unsigned int *) (i)) = 0x00;         : Erases from i to i + 2 KB.
  while (FMC_STAT<WBUSY[0]> == 1);}                : Checks that FMC is not busy.

FMC_CTRL<SERS[2]> = 0;                             : Disables 2 KB-sector erase mode.
```

<Example 3> Read protection disablement

```
while (FMC_STAT<WBUSY[0]> == 1);                   : Checks that FMC is not busy.

FMC_CTRL<CERS[3]> = 1;                             : Enables chip erase mode.

  (*(volatile unsigned int *) (0x0)) = 0x00;       : Executes chip erase.
  while (FMC_READPROT<CERSD[24]> == 1);            : Raises the chip erase done flag.

FMC_CTRL<CERS[3]> = 0;                             : Disables chip erase mode.
  while (FMC_STAT<WBUSY[0]> == 1);                 : Checks that FMC is not busy.

FMC_CTRL<PERS[1]> = 1;                             : Enables page erase mode.

(*(volatile unsigned int *) (0x0F000000)) = 0x00;  : Executes erase in the OTP0 area.

FMC_READPROT<RPROT[7:0]> = 0xFF;                   : Disables read protection.
```

7 Direct Memory Access Controller (DMAC)

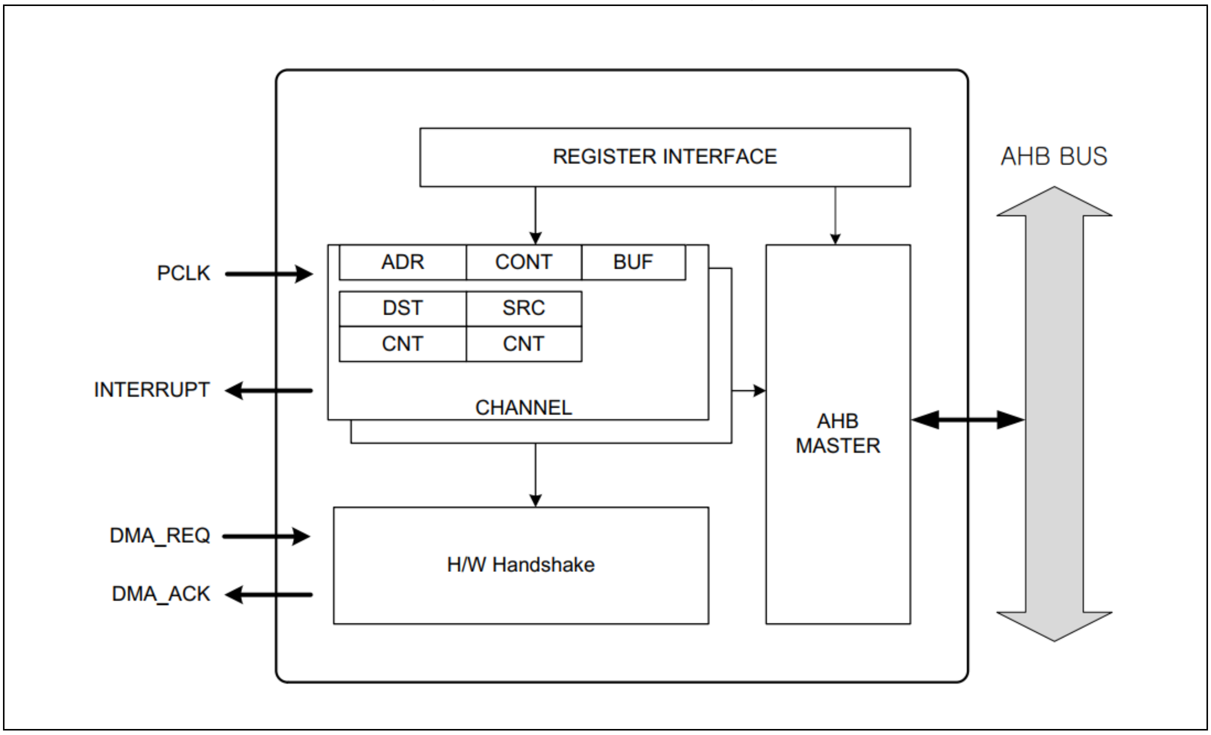
Direct Memory Access (DMA) controller is used for high-speed data transfers between peripherals and memories. The DMA enables quick data transfers having memory to memory copying or moving of data within memory.

- 8 channels
- Only single-ended signaling supported
- 8-/16-/32-bit data transfers supported
- Various buffers with the same size supported
- DMA transfers are triggered through peripheral interrupts

7.1 DMAC block diagram

In this section, DMAC block diagram is introduced in Figure 42.

Figure 42. DMAC Block Diagram



7.2 Registers

Base address of DMAC is introduced in the followings:

Table 33. Base Address of DMAC

Name	Base address
DMACH0	0x4000_0400
DMACH1	0x4000_0410
DMACH2	0x4000_0420
DMACH3	0x4000_0430
DMACH4	0x4000_0440
DMACH5	0x4000_0450
DMACH6	0x4000_0460
DMACH7	0x4000_0470

Table 34. DMAC Register Map

Name	Offset	Type	Description	Reset value	Reference
DMAn_CR	0x0000	RW	DMA Channel n Control Register	0x0000_0000	7.2.1
DMAn_SR	0x0004	RW	DMA Channel n Status Register	0x0000_0080	7.2.2
DMAn_PAR	0x0008	RW	DMA Channel n Peripheral Address	0x4000_0000	7.2.3
DMAn_MAR	0x000C	RW	DMA Channel n Memory Address	0x2000_0000	7.2.4

7.2.1 DMAn_CR: DMA controller register

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

DMA0_CR=0x4000_0400, DMA1_CR=0x4000_0410, DMA2_CR=0x4000_0420, DMA3_CR=0x4000_0430
DMA4_CR=0x4000_0440, DMA5_CR=0x4000_0450, DMA6_CR=0x4000_0460, DMA7_CR=0x4000_0470

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	-	0x0	-	00	0	-
-	RW	-	RW	-	RW	RW	-

Bits	Name	Value	Description
27 16	TRANSCNT		Number of DMA transfers Before enabling DMA transfer, the number of transfers must be written to TRANSCNT.
		0	All DMA transfers have been completed.
		N	There are N transfers remaining.
11 8	PERISEL		Peripheral selection The selected peripheral is connected to the DMA channel. PERISEL must be written with the number representing the peripheral to be connected to the DMA interface.
		N	Selects the peripheral to use. Refer to Table 35 of DMA peripheral numbers.
3 2	SIZE		Bus transfer size
		00	Sets the DMA transfer size to 1 byte.
		01	Sets the DMA transfer size to half-word size.
		10	Sets the DMA transfer size to one-word size.
		11	Holds off the size setting.
1	DIR		Transfer direction selection
		0	Memory -> peripheral (TX)
		1	Peripheral -> memory (RX)

NOTE: A DMA channel will be connected with selected peripheral. Table 35 shows peripheral selection numbers. This PERISEL field should be set with proper number of peripheral which will be connected with DMA interface.

Table 35. DMAC PERISEL Selection

PERISEL[11:8]	Associate peripheral	PERISEL[11:8]	Associate peripheral
0	CHANNEL IDLE	8	UART3 TX
1	UART0 RX	9	SPI0 RX
2	UART0 TX	10	SPI0 TX
3	UART1 RX	11	SPI1 RX
4	UART1 TX	12	SPI1 TX
5	UART2 RX	13	ADC0 RX
6	UART2 TX	14	ADC1 RX
7	UART3 RX	15	CRC Tx

NOTE: 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 DMAn_SR: DMA status register

DMAn_SR is a 32-bit register. It shows the current status of the DMA controller and whether the DMA channel is enabled or disabled.

DMA0_SR=0x4000_0404, DMA1_SR=0x4000_0414, DMA2_SR=0x4000_0424, DMA3_SR=0x4000_0434
DMA4_SR=0x4000_0444, DMA5_SR=0x4000_0454, DMA6_SR=0x4000_0464, DMA7_SR=0x4000_0474

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	DMARC	Reserved	DMAEN
-	1	-	0	-	1
-	RO	-	WC	-	RW

7	EOT	End of transfers	
		0	There are remaining transfers. TRANSCMT has a value other than zero.
		1	All data has been transferred. The TRANSCMT value is zero.
4	DMARC	DMA request clear (Auto clear)	
		0	Not use
		1	DMA Initialization
0	DMAEN	DMA enablement	
		0	The DMA is inactive or disabled.
		1	The DMA is active or enabled.

NOTE: When DMA Enable is enabled, it should be used after initializing DMA Request Clear bit to '1'.

```
DMAn_SR.DMARC[4] = "1" // DMA Initialization
DMAn_SR.DMAEN[0] = "1" // DMA enablement
```

7.2.3 DMAn_PAR: DMA n peripheral device address register

DMAn_PAR shows the address value of the connected peripheral.

DMA0_PAR=0x4000_0408, DMA1_PAR=0x4000_0418, DMA2_PAR=0x4000_0428
DMA3_PAR=0x4000_0438, DMA4_PAR=0x4000_0448, DMA5_PAR=0x4000_0458
DMA6_PAR=0x4000_0468, DMA7_PAR=0x4000_0478

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	PAR
0x4000	0x0000
RW	RO

31 0	PAR	Address of the transfer/receive buffer's target peripheral The address does not change until the ongoing transfer is completed.

7.2.4 DMAn_MAR: DMA n memory address register

DMAn_MAR shows the DMA transfer’s target memory address.

DMA0_MAR=0x4000_040C, DMA1_MAR=0x4000_041C, DMA2_MAR=0x4000_042C
DMA3_MAR=0x4000_043C , DMA4_MAR=0x4000_044C, DMA5_MAR=0x4000_045C
DMA6_MAR=0x4000_046C, DMA7_MAR=0x4000_047C

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	
RW		RW	

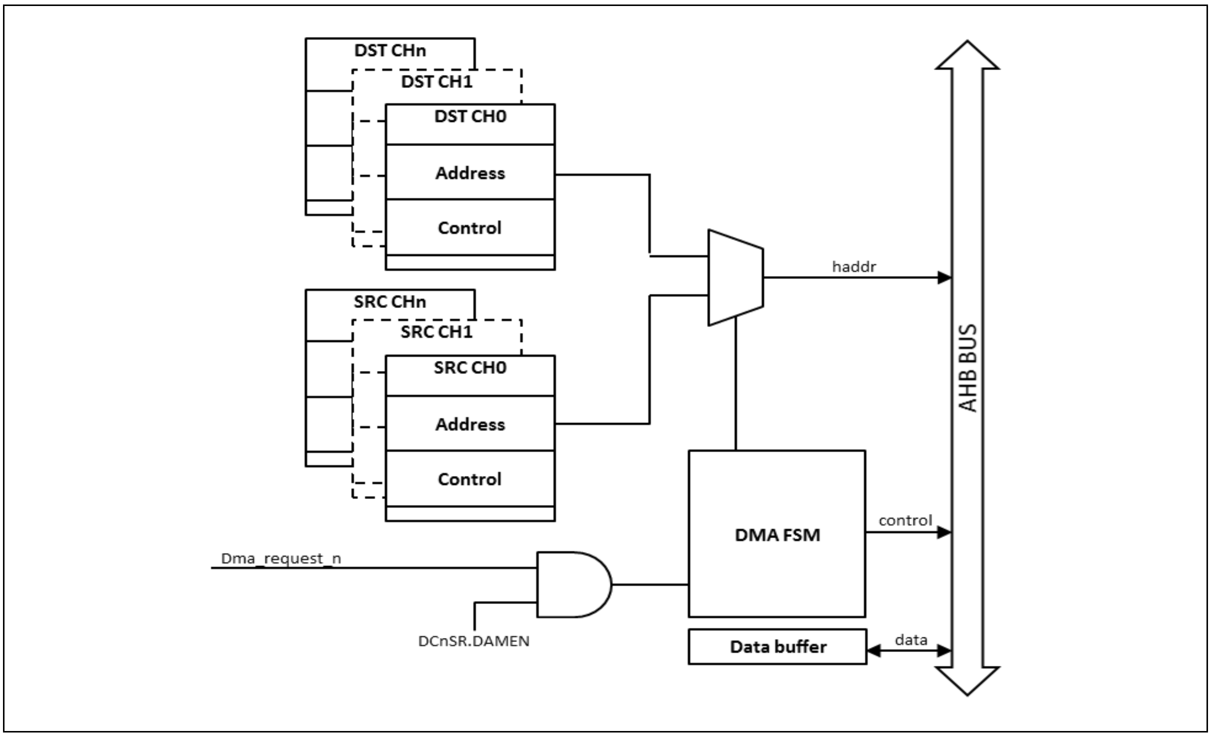
31 0	MAR	Target memory address of the data transfer On the completion of each transfer, the address automatically increases, depending on the setting of the SIZE bits.

7.3 Functional description

The DMA controller directly transfers data to memories by sharing the AHB bus with the CPU core. The AHB shares two masters operating in a round-robin fashion. Therefore, the DMA controller shares only half the system bandwidth with the CPU core.

The DMA controller is triggered only by requests made by peripherals. Once a peripheral requests a transfer to the DMA controller, the connected channel becomes enabled. Then, the bus is accessed to transfer the requested data from the memory buffer to the peripheral's data buffer (or the other way around).

Figure 43. DMA Controller Block Diagram



The DMA controller operates in the following sequence:

1. The programmer sets the address of the peripheral to use in the DMAn_PAR register and the memory address in the DMAn_MAR register.
2. In the DMAn_CR register, the programmer sets the DMA transfer count (0–4095), transferring direction, and bus transfer size (8-/16-/32-bit).
3. In the DMAn_SR register, the programmer sets the DMEAN bit to 1 to enable the DMA channel.
4. A DMA request is made by the peripheral set in DMAn_CR's PERISEL bits and this triggers the requested peripheral channel to become active.

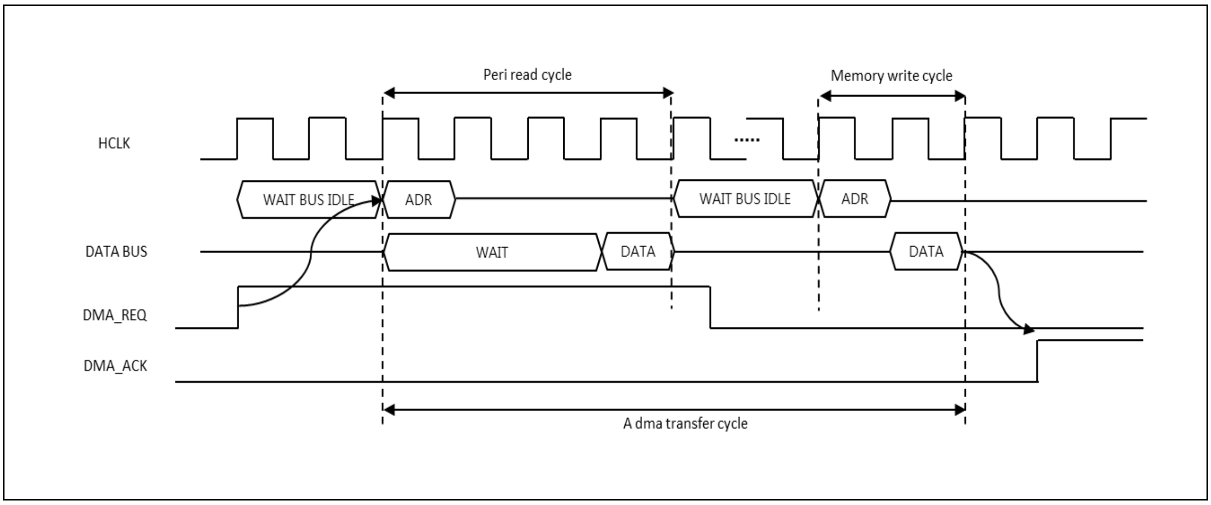
5. Data received from the source DMA address is read and stored in the internal buffer. The DMA operation writes this data to the target address.
6. Each time data is written to the target address, the DMA transfer count decreases by 1. When the DMA transfer count reaches 0, DMAn_SR's EOT bit is set to 1. This signals an interrupt to the connected peripheral.

NOTE: DMA itself has no interrupt sources; instead, each peripheral has flag bits that display the status of the connected DMA channel's transfer interrupts.

7.3.1 DMA transfer timing from a peripheral to a memory

The diagram below shows the functional timing of the DMA controller. A transfer request from a peripheral is internally held off while the AHB calls the address of the data read transfer source. The data read from the source address is stored in the internal buffer. Then, once the AHB becomes available, the stored data is transferred to the target address. As shown in the diagram below, a waiting time of four clock cycles is required until the peripheral is accessed. If the bus is occupied by another master, additional wait cycles will be taken.

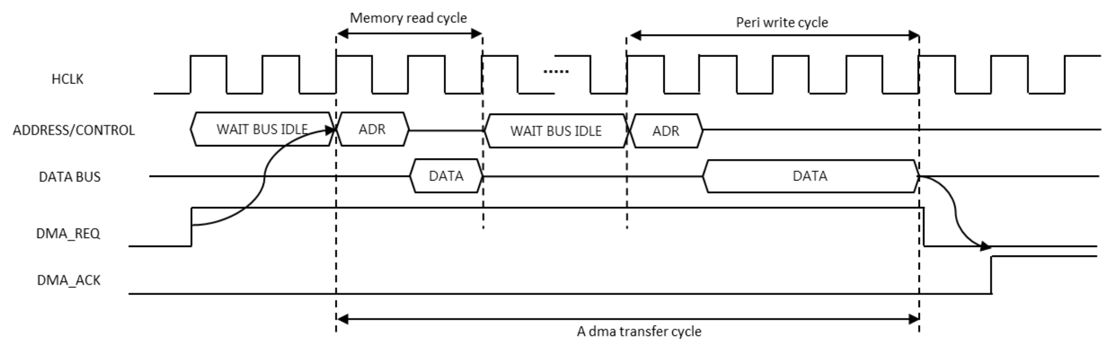
Figure 44. Diagram of DMA Transfer Timing from a Peripheral to a Memory



7.3.2 DMA transfer timing from a memory to a peripheral

The diagram below depicts the timing of a DMA transfer made from a memory to a peripheral. A waiting time of four clock cycles is required until the peripheral is accessed. If the bus is occupied by another master, additional wait cycles will be taken.

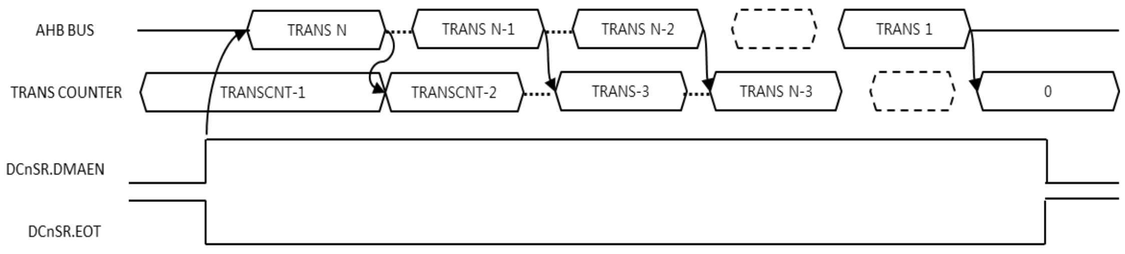
Figure 45. Diagram of DMA Transfer Timing from a Memory to a Peripheral



7.3.3 DMA transfer

The diagram below depicts N number of DMA transfers in the queue as an example. DMA transferring starts when DMAn_SR's DMAEN bit is set to 1. And the bit is cleared to 0 when all transfers have been completed.

Figure 46. N Times of DMA Transfers



7.3.4 Setting examples

<Example 1> DMA0 data transfer from peripheral (SPI0, 0x40009004) to memory (0x20001000)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                  : Unlocks the SCU registers.
SCU_PER1<DMA[4]> = “1”                            : Enables the DMA peripheral.
SCU_PCER1<DMA[4]> = “1”                           : Enables the DMA peripheral clock.

DMA0_CR<TRANSCNT[27:16]> = “00000000 0001”        : Enters 1 as the number of DMA transfers.
DMA0_CR<PERISEL[11:8]> = “1001”                   : Selects a peripheral (SPI Rx).
DMA0_CR<SIZE[3:2]> = “10”                         : Sets the transfer size to word size (32 bits).
DMA0_CR<DIR[1]> = “1”                             : Sets the transfer direction to peripheral ➔ memory.
DMA0_PAR<PAR[15:0]> = “10010000 00000100”         : Enters the address of the peripheral (SPI0) to which data will
                                                  be transferred (0x40009004).
DMA0_MAR<MAR[15:0]>=”00010000 00000000”           :  Enters the memory address from which data will be loaded
                                                  (0x20001000).
DMA0_SR<DMARC[4]> = “1”                           : DMA request clear.
DMA0_SR<DMAEN[0]> = “1”                           : Enables DMA transfer.
```

<Example 2> DMA0 data transfer from memory (0x20001000) to peripheral (SPI0, 0x40009004)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                  : Unlocks the SCU registers.
SCU_PER1<DMA[4]> = “1”                            : Enables the DMA peripheral.
SCU_PCER1<DMA[4]> = “1”                           : Enables the DMA peripheral clock.

DMA0_CR<TRANSCNT[27:16]> = “00000000 0001”        : Enters 1 as the number of DMA transfers.
DMA0_CR<PERISEL[11:8]> = “1010”                   : Selects a peripheral (SPI Tx).
DMA0_CR<SIZE[3:2]> = “10”                         : Sets the transfer size to word size (32 bits).
DMA0_CR<DIR[1]> = “0”                             : Sets the transfer direction as memory peripheral.

DMA0_PAR<PAR[15:0]> = “10010000 00000100”         : Enters the address of the peripheral (SPI0) from which data will
                                                  be loaded (0x40009004).

DMA0_MAR<MAR[15:0]>=”00010000 00000000”           :  Enters the memory address to which data will be transferred
                                                  (0x20001000).
DMA0_SR<DMARC[4]> = “1”                           : DMA request clear.
DMA0_SR<DMAEN[0]> = “1”                           : Enables DMA transfer.
```

8 Watchdog Timer (WDT)

Watchdog Timer (WDT) monitors the operation of the MCU and is typically used to detect software errors. When the MCU becomes uncontrollable due to a malfunction, the WDT resets the MCU to recover it.

A33M11x series has one WDT module built in, which functions as a 32-bit down-counter. Once the WDT counts down to zero while set as a reset source, the MCU gets reset. When it is not used to monitor the MCU, it can be used as a cycle timer along with an interrupt.

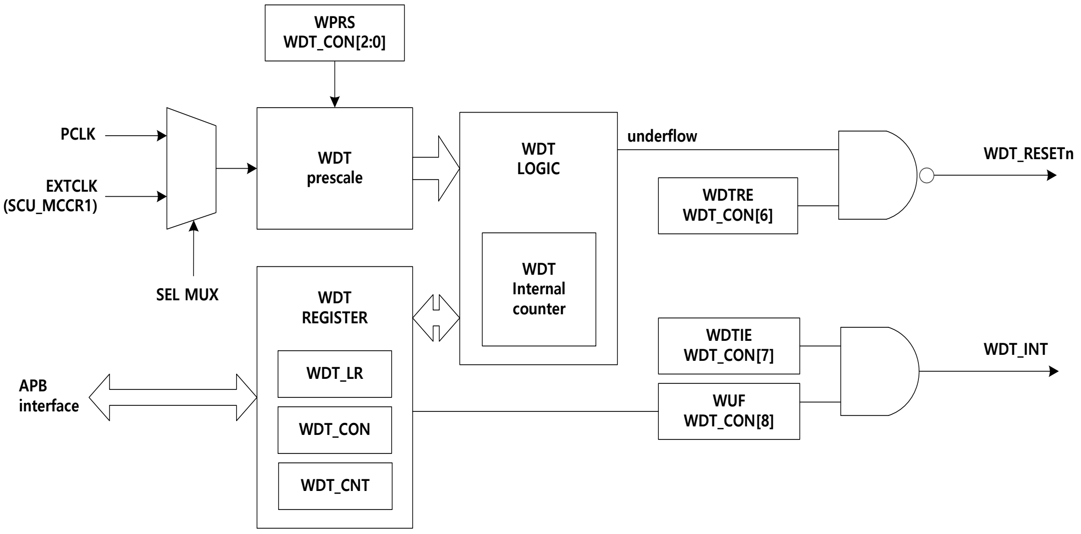
The WDT of A33M11x series features the followings:

- A 32-bit down-counter
- WDT underflow reset supported
- Cycle timer and underflow interrupt supported
- WDT input clock sources selectable
  - PCLK
  - Clock sources selectable with the setting of SCU_MCCR1[26:24]: LSI, MCLK, HSI, HSE, PLL
- Eight-level prescalers for the WDT clock
- The user can set whether to enable or disable the WDT counter in debug mode

8.1 WDT block diagram

In this section, WDT block diagram is introduced in Figure 47.

Figure 47. WDT Block Diagram



8.2 Registers

Initial watchdog time-out period is set to 2,000-milisecond. Base address of WDT is introduced in the followings:

Table 36. Base Address of WDT

Name	Base address
WDT	0x4000_0200

Table 37. WDT Register Map

Name	Offset	Type	Description	Reset value	Reference
WDT_LR	0x0000	WO	WDT load register	0x0000_0000	8.2.1
WDT_CNT	0x0004	RO	WDT current count register	0x0000_7A12	8.2.2
WDT_CON	0x0008	RW	WDT control register	0x0000_805C	8.2.3
WDT_AEN	0x00F0	WO	WDT access enable register	0x0000_0000	8.2.4

8.2.1 WDT_LR: watchdog timer load register

WDT_LR is used to update the value of the WDT_CNT register. To change WDT_CNT's value, two conditions must be satisfied: WDT_CON's WEN bit must be set to 1, and WDT_LR must be written to. If WDT_CON's WEN bit is set to 0, the value written to the WDT_LR will remain unrepresented in WDT_CNT until the WEN bit is changed to 1.

If the WDT is being used as a reset source, WDT_LR must be written to before the WDT_CNT value becomes 0 to prevent a reset.

The WDT triggers an event at the moment when the WDT_CNT value is changed from 1 to 0. Therefore, when the WDT is used in reset mode, the WDT's count value is written to WDT_LR as it is; whereas when the WDT is used in interrupt mode, 1 must be subtracted from the count value before WDT_LR is written to. At least five WDT counter clock cycles are required to update WDT_CNT with the WDT_LR value.

WDT_LR=0x4000_0200

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
WDTLR			
0x0000_0000			
RW			

31 0	WDTLR	WDT load value register If the WDTEN remains at 1, the LR register will be updated with the CNT value.

8.2.2 WDT_CNT: watchdog current count register

WDT_CNT is a 32-bit down-counter that shows the WDT's current value. It is a read-only register. Its value can be changed by writing to WDT_LR while the set value of WDT_CON's WEN bit is 1.

If the WDT is used as a reset source, a reset occurs when the WDT_CNT value becomes 0. If it is used as a cycle timer, an interrupt occurs when the WDT_CNT value becomes 0.

To use the WDT as a reset source, both WDT_CON's WRE bit and SCU_RSER's WDTRST bit must be set to 1.

Thus, when the WDT's count value reaches 0, an interrupt or reset is triggered.

WDT_CNT=0x4000_0204

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
WDTCNT																															
0x0000_7A12																															
RO																															

31 0	WDTCNT	WDT current count register The 32-bit down-counter counts down from the value written to WDT_LR.

8.2.3 WDT_CON: watchdog control register

The MCU’s WDT module must be set appropriately before it is enabled. The WDT module can be programmed to trigger a core reset event or interrupt signal. Instead of being used as a reset source or an interrupt source, the WDT can also function as a countdown timer staring from the set down-counter value. The WUF bit is a flag that is set when the WDT counts down to zero.

WDT_CON=0x4000_0208

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	WDBG	Reserved	WUF	WDTIE	WDTRE	Reserved	WDTEN	CKSEL	WPRS
-	1	-	0	0	1	-	1	1	100
-	RW	-	RW	RW	RW	RW	RW	RW	RW

15	WDBG	Whether to enable or disable the WDT in debug mode	
		0	Enables the WDT in debug mode (STOP).
		1	Disables the WDT in debug mode (STOP).
8	WUF	WDT underflow flag (The bit is cleared when WDT_LR is written to.)	
		0	There is no underflow.
		1	Underflow is pending.
7	WDTIE	Whether to enable or disable the WDT counter underflow interrupt	
		0	Disables the interrupt.
		1	Enables the interrupt.
6	WDTRE	Whether to enable or disable the WDT counter underflow reset	
		0	Disables the WDT counter underflow reset.
		1	Enables the WDT counter underflow reset.
4	WDTEN	Whether to enable or disable the WDT counter	
		0	Disables the WDT counter.
		1	Enables the WDT counter.
3	CKSEL	WDTCLKIN clock source selection	
		0	PCLK
		1	External clock (MCCR1)
2 0	WPRS[2:0]	Counter clock prescaler WDTCLK = WDTCLKIN/WPRS	
		000	WDTCLKIN
		001	WDTCLKIN / 4
		010	WDTCLKIN / 8
		011	WDTCLKIN / 16
		100	WDTCLKIN / 32
		101	WDTCLKIN / 64
		110	WDTCLKIN / 128
		111	WDTCLKIN / 256

8.2.4 WDT_AEN: watchdog access enable register

The WDT_AEN register determines whether or not to allow changes to the settings of all WDT registers. In addition, the register functions to reload an input by the user instantly when it is entered.

WDT_AEN=0x4000_02F0

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td>ENS</td><td colspan="16">AEN</td></tr>
<tr><td colspan="15">-</td><td>0</td><td colspan="16">--</td></tr>
<tr><td colspan="15"></td><td>RO</td><td colspan="16">WO</td></tr>
</table>

16	ENS	Whether the register is enabled or disabled	
		0	Disables AEN.
		1	Enables AEN (WDT registers can be accessed).
15 0	AEN	Writing 0xA55A to the bit field enables writing new values to WDT registers. After this, write a different value to this bit field to protect the WDT registers against being updated with new values. Additionally, writing 0x555A to the bit field triggers an instant reload. When reloading, the LR register should be initialized before use.	

NOTE: Example code for using AEN

```
WDT_AEN=0xA55A;       // Enables AEN.
                  // LR and CON, etc. become settable.
WDT_AEN=0;            // Disables AEN.
```

To use the immediate reload feature, a value greater than 0 must be set in the WDT_LR register.

```
WDT_AEN=0xA55A;  // Enable AEN
WDT_LR=0x7A12;   // Set timer load value
WDT_AEN=0;       // Disable AEN
WDT_AEN=0x555A;  // Triggers an instant reload.
```

8.3 Functional description

8.3.1 WDT down-counter control

Once the WDTEN (WDT_CON[4]) bit is set to 1, WDT_CNT's 32-bit down-counter starts counting down. To prevent an instant reset or interrupt, WDT_CNT must be written to.

To change the WDT_CNT value, a value larger than 0 must be written to WDT_LR while the set value of WDT_CON's WDTEN bit is 1. If the active down-counter is reloaded by WDT_LR, the counter value will be reset. In this case, the down-counter's value must be larger than zero.

8.3.2 WDT reset mode

To use the WDT as a reset source, both WDT_CON's WDTRE bit and SCU_RSER's WDTRST bit must be set to 1. At the moment the down-counter value changes from 1 to 0, a WDT reset occurs with a software error detected.

8.3.3 WDT interrupt mode

The WDT interrupt can be used to perform safety tasks or data logging before triggering an actual reset.

The WDT interrupt is enabled by setting WDT_CON's WDTIE bit to 1. At the moment the WDT_CNT down-counter value changes from 1 to 0, the interrupt is triggered and WDT_CON's WUF bit is set to 1. To clear the WUF bit, you must write a value other than 0 to WDT_CNT by writing to WDT_LR.

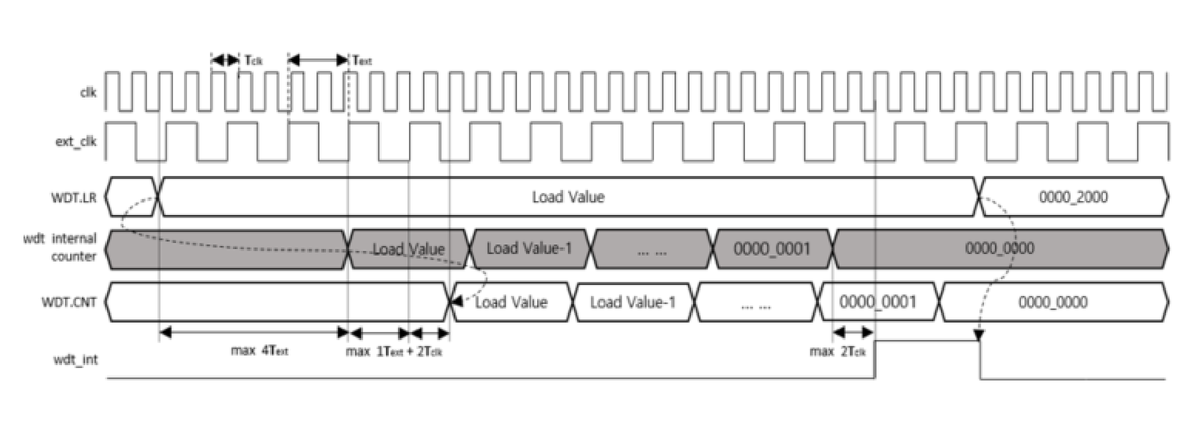
8.3.4 Using the WDT in debug mode

By setting WDT_CON's WDBG bit to 1, you can stop the WDT counter in debug mode. If you press "stop" for a running WDT counter in debug mode, the counter value will stop changing. If you press "run," the counter value will resume decrementing by 1.

Conversely, if the WDBG bit is set to 0, the WDT counter will continue running in debug mode without being stopped. Once you press "stop" in debug mode, the counter value will seem unchanged; however, if you run the counter again, you will find the counter value having decreased, not stopped.

8.3.5 Timing diagram

Figure 48. Diagram of Interrupt Mode Operation Timing with External Clock as WDT Clock



In WDT interrupt mode, to prevent another occurrence of the WDT interrupt shortly after an occurrence of WDT underflow, a certain counter value is reloaded. This reload is enabled only when the WDT counter is set to run in interrupt mode.

It takes five cycles that the counter value is updated with a new load value. The WDT interrupt signal and the CNT value data can be delayed up to two system bus clock cycles if the WDT runs in a synchronous logic circuit.

8.3.6 Prescaler table

The WDT includes a programmable 32-bit down-counter prescaler that enables you to set the timeout period in various ways.

As the WDT’s clock source, either the PCLK or an external clock whose frequency is multiplied by 1/5 can be used. An external clock source can be used by setting the CON register’s CKSEL (3rd bit) to 1 and selecting the external clock source in MCCR1.

To make a WDT counter base clock, the user can control the 3-bit prescaler by setting the WDT_CON register’s WPRS bits [2:0]. The maximum frequency prescaler value for a clock source is “1/256.”

The table below lists prescaled frequencies for each WDT counter clock source.

Selectable clock source (LSI freq to PLL freq) 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 38. Prescaled WDT Counter Clock Frequencies

Clock source	WDTCLKIN	WDTCLKIN/4	WDTCLKIN/16	WDTCLKIN/64	WDTCLKIN/256
LSI	500KHz	125KHz	31.25KHz	7.8125KHz	1.953125KHz
MCLK	MCLK (BUS CLK)	MCLK/4	MCLK/16	MCLK/64	MCLK/256
HSI32	32MHz	8MHz	2MHz	500KHz	125KHz
HSE	XTAL	XTAL/4	XTAL/16	XTAL/64	XTAL/256

8.3.7 Setting examples

<Example 1> WDT cycle interrupt mode - PCLK 8MHz, 2 s period

```
WDT_AEN<AEN[15:0]> = “10100101_01011010”            : Enables WDT access (0xA55A).
WDT_CON<WDBG[15]> = “0”                             : Initializes the WDT control register.
WDT_CON<WUF[8]> = “0”
WDT_CON<WDTIE[7]> = “0”
WDT_CON<WDTRE[6]> = “0”
WDT_CON<WDTEN[4]> = “0”
WDT_CON<CKSEL[3]> = “0”
WDT_CON<WPRS[2:0]> = “000”
WDT_LR<WDTLR[31:0]> = “00000000 00000000 00000000   : Initializes the WDT load register.
00000000”

WDT_CON<WDTIE[7]> = “1”                             : Enables the WDT interrupt.
WDT_CON<WDTRE[6]> = “0”                             : Disables the WDT reset.
WDT_CON<CKSEL[3]> = “0”                             : Sets the PCLK as the WDT clock source (8MHz).
WDT_CON<WPRS[2:0]>=”001”                            : Sets the WDT clock prescaler value.
WDT_LR<WDTLR[31:0]>                                 : Sets the period to 2 seconds (when 1 tick = (1/8) us).
= ”00000000_00111101_00001001_00000000”                        → 2s / (1/8)us = 4,000,000

WDT_CON<WDTEN[4]> = ”1”                             : Enables the WDT.

NVICIP[6]<PRI_6[23:16]> = “00110000”                : Sets the NVIC WDT priority level.
NVICISER[0]<SETPEND[31:0]>                          : Enables the NVIC WDT interrupt.
= “00000000_00000000_00000000_01000000”
```

9 16-bit timer

A33M11x series has eight channels of 16-bit timers built in. These 16-bit timers support four operating modes: periodic, PWM, One-shot, and capture modes. As the input clock source to the 16-bit timers, either a divided PCLK or an external clock can be used. Additionally, an internal 10-bit prescaler allows the user to generate various timer base clocks.

Interrupts can be triggered at regular intervals when a timer is used in periodic mode. The user can set the period and duty to form a PWM signal to be used in PWM mode. In One-shot and PWM modes, the timer can generate one PWM waveform. In capture mode, the external input signal's pulse intervals can be measured based the preset conditions. Moreover, the timer can export signals to other devices to control them. These timers are primarily used as periodic tick timers or wake-up sources.

16-bit timer of A33M1x series features the followings:

1. 16-bit up-counter timers
2. Four operating modes:
   - Periodic timer mode
   - One-shot timer mode
   - PWM mode
   - Capture mode
3. Various interrupts:
   - Match/overflow interrupts
4. Timer input clock sources selectable:
   - Four PCLK prescaler levels (1/2, 1/4, 1/16, 1/64)
   - Clock sources selectable with the setting of SCU_MCCR3: LSI, MCLK, HSI, HSE, and PLL
   - Timer clock source by an input to port TnIO
5. 10-bit prescaler built in to support the timer input clock
6. PWM synchronization
   - Start delay and clear synchronization

Table 39 introduces pins assigned for 16-bit timer.

Table 39. Pin Assignment of 16-bit Timer: External Pins

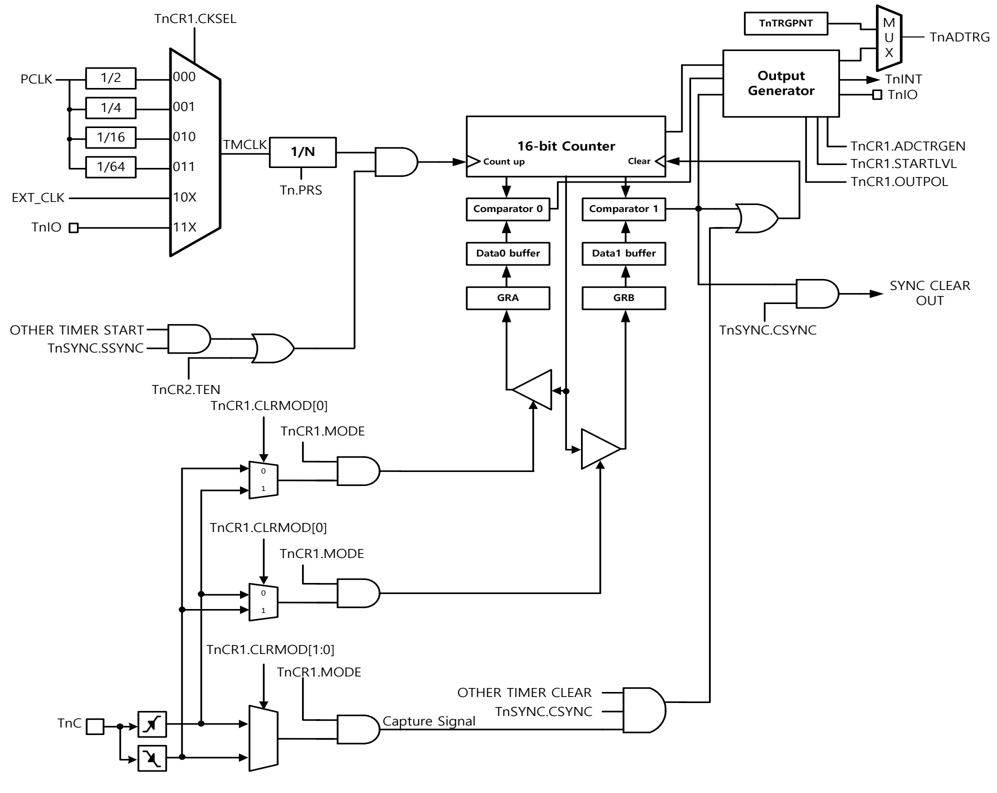
Pin name	Type	Description	Supported Packages	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
T0IO	IO	Timer0 capture input signal/external clock input Timer0 timer/PWM/One-shot output	O	O
T1IO	IO	Timer1 capture input signal/external clock input Timer1 timer/PWM/One-shot output	O	O
T2IO	IO	Timer2 capture input signal/external clock input Timer2 timer/PWM/One-shot output	O	O
T3IO	IO	Timer3 capture input signal/external clock input Timer3 timer/PWM/One-shot output	O	O
T4IO	IO	Timer4 capture input signal/external clock input Timer4 timer/PWM/One-shot output	O	O
T5IO	IO	Timer5 capture input signal/external clock input Timer5 timer/PWM/One-shot output	O	O
T6IO	IO	Timer6 capture input signal/external clock input Timer6 timer/PWM/One-shot output	O	O
T7IO	IO	Timer7 capture input signal/external clock input Timer7 timer/PWM/One-shot output	O	O

NOTE: If the package is reduced, the timer function can be used internally, but the external pin cannot be used.

9.1 16-bit timer block diagram

In this section, 16-bit timer is described in a block diagram in Figure 49.

Figure 49. 16-bit Timer Block Diagram



Checked boxes can be selected by group
(TIMER A : T0~T3, TIMER B : T4~T7)

9.2 Registers

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

Table 40. Base Address of 16-bit Timer

Name	Base address
TIMER0	0x4000_3000
TIMER1	0x4000_3040
TIMER2	0x4000_3080
TIMER3	0x4000_30C0
TIMER4	0x4000_3100
TIMER5	0x4000_3140
TIMER6	0x4000_3180
TIMER7	0x4000_31C0

Table 41. TIMER Register Map

Name	Offset	Type	Description	Reset value	Reference
TIMERn_CR1	0x0000	RW	Timer n control register 1	0x0000_0000	9.2.1
TIMERn_CR2	0x0004	RW	Timer n control register 2	0x0000_0000	9.2.2
TIMERn_PRS	0x0008	RW	Timer n prescaler register	0x0000_0000	9.2.3
TIMERn_GRA	0x000C	RW	Timer n general data register A	0x0000_0000	9.2.4
TIMERn_GRB	0x0010	RW	Timer n general data register B	0x0000_0000	9.2.5
TIMERn_CNT	0x0014	RW	Timer n counter register	0x0000_0000	9.2.6
TIMERn_SR	0x0018	RC	Timer n status register	0x0000_0000	9.2.7
TIMERn_IER	0x001C	RW	Timer n interrupt enable register	0x0000_0000	9.2.8
TIMERn_TRGPNT	0x0020	RW	Timer n trigger point register	0x0000_0000	9.2.9
TIMERn_SYNC	0x0024	RW	Timer n sync configuration register	0x0000_0000	9.2.10

NOTE: n = 0, 1, 2, 3, 4, 5, 6 and 7

9.2.1 TIMERn_CR1: timer/counter n control register 1

TIMERn_CR1 is a 32-bit register. The timer module must be set appropriately before running the timer. After the intended use of the timer is specified, the timer can be set in CR1. After this register is set, you can enable or disable the timer by setting CR2.

TIMER0_CR1=0x4000_3000, TIMER1_CR1=0x4000_3040, TIMER2_CR1=0x4000_3080
TIMER3_CR1=0x4000_30C0, TIMER4_CR1=0x4000_3100, TIMER5_CR1=0x4000_3140
TIMER6_CR1=0x4000_3180, TIMER7_CR1=0x4000_31C0

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	TRGMOD	UAO	OUTPOL	IOSEL	Reserved	ADCTRGEN	STARTLVL	CKSEL	CLRMOD	MODE
-	00	0	0	0	-	0	0	000	00	00
-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW

15 14	TRGMOD	ADC trigger mode selection	
		0X	Selects GRA value trigger mode (normal mode).
		10	Selects TRGPNT value trigger mode.
		11	Triggers both GRA and TRGPNT values.
13	UAO	GRA/GRB update mode selection	
		0	The value written to GRA or GRB is applied after the current period.
		1	The value written to GRA or GRB is applied in the current period.
12	OUTPOL	Timer output polarity	
		0	General output
		1	General output inversion
11	IOSEL	TnIO pin configuration	
		0	Sets the pin as an input port (TnIO).
		1	Sets the pin as an output port (TnIO).
8	ADCTRGEN	Whether or not to use the timer as an ADC trigger source	
		0	Does not use the timer as an ADC trigger source.
		1	Uses the timer as an ADC trigger source.
7	STARTLVL	Starting output value in periodic/PWM/One-shot modes	
		0	Sets the starting output value to “L.”
		1	Sets the starting output value to “H.”
6 4	CKSEL[2:0]	Counter clock source selection	
		000	PCLK/2
		001	PCLK/4
		010	PCLK/16
		011	PCLK/64
		10X	EXT0 (MCCR3)
		11X	Input to pin TnIO
3 2	CLRMOD	Clear mode selection in capture mode	
		00	Rising-edge clear mode
		01	Falling-edge clear mode
		10	Both-edge clear mode
		11	Disables clearing.
1 0	MODE	Timer operation mode control	
		00	Normal periodic mode

	01	PWM mode
	10	One-shot mode
	11	Capture mode

9.2.2 TIMERn_CR2: timer/counter n control register 2

TIMERn_CR2 is a 32-bit register and is used to control the timer's operation. Before you start the timer, you must set CR2's TCLR bit to 1 to clear the timer count register.

TIMER0_CR2=0x4000_3004, TIMER1_CR2=0x4000_3044, TIMER2_CR2=0x4000_3084
TIMER3_CR2=0x4000_30C4, TIMER4_CR2=0x4000_3104, TIMER5_CR2=0x4000_3144
TIMER6_CR2=0x4000_3184, TIMER7_CR2=0x4000_31C4

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																														TCLR	TEN
-																														0	0
-																														WO	RW

1	TCLR	Clearing of the timer count register	
		0	Does not clear the register.
		1	Initializes the timer. Writing a 1 to TCLR clears the counter register. (* Write-only bit)
0	TEN	Whether to enable or disable the timer	
		0	Disables the timer.
		1	Enables the timer.

NOTE: You must set the TCLR bit to 1 before starting the timer to prevent a time error.

9.2.3 TIMERn_PRS: timer/counter n prescaler register

TIMERn_PRS is used to set the timer input frequency divider. It is 10-bit wide. You can generate precise and varied timer base clocks by applying the prescaler to the timer clock source that has been selected in TIMERn_CR1.

TIMER0_PRS=0x4000_3008, TIMER1_PRS=0x4000_3048, TIMER2_PRS=0x4000_3088
TIMER3_PRS=0x4000_30C8, TIMER4_PRS=0x4000_3104, TIMER5_PRS=0x4000_3144
TIMER6_PRS=0x4000_3184, TIMER7_PRS=0x4000_31C4

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																						PRS									
-																						000									
-																						RW									

0	PRS	Prescaler value for the counter clock TCLK = CLOCK_IN/(PRS+1) (CLOCK_IN = timer input clock selected by the setting of CR1's CKSEL bit)

9.2.4 TIMERn_GRA: timer/counter n general data register A

TIMERn_GRA is a 32-bit register.

TIMER0_GRA=0x4000_300C, TIMER1_GRA=0x4000_304C, TIMER2_GRA=0x4000_308C
TIMER3_GRA=0x4000_30CC, TIMER4_GRA=0x4000_310C, TIMER5_GRA=0x4000_314C
TIMER6_GRA=0x4000_318C, TIMER7_GRA=0x4000_31CC

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		GRA	
-		0x0000	
-		RW	

15 0	GRA	Timer n general register A This register is used for different purposes depending on the operating mode.
		Period/PWM/One-shot Modes The target count value is stored in the register. In periodic mode, the GRA value must be greater than 0 and less than or equal to the GRB value. If the counter value matches the GRA value, the counter is cleared to restart or stop running. When the timer is cleared, TOUT outputs the TSTRT value. If the counter value matches the register's value, the GRA interrupt is triggered. When the GRA interrupt is triggered or clearing is demanded, the counter value is copied to internal data buffer 0. In PWM mode, the GRA value represents the duty value.
		Capture mode - In rising-edge clear mode, the register stores the counter value captured on the falling edge of the signal at port TnIO. - In falling-edge clear mode, the register stores the counter value captured on the rising edge of the signal at port TnIO.

9.2.5 TIMERn_GRB: timer/counter n general data register B

TIMERn_GRB is a 32-bit register.

TIMER0_GRB=0x4000_3010, TIMER1_GRB=0x4000_3050, TIMER2_GRB=0x4000_3090
TIMER3_GRB=0x4000_30D0, TIMER4_GRB=0x4000_3110, TIMER5_GRB=0x4000_3150
TIMER6_GRB=0x4000_3190, TIMER7_GRB=0x4000_31D0

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		GRB	
-		0x0000	
-		RW	

15 0	GRB	Timer n general register B This register is used for different purposes depending on the operating mode.
		Periodic mode In periodic mode, GRB value is the period.
		PWM/One-shot mode The target count value is stored in the register. If the counter value matches the register's value, the GRB interrupt is triggered. In PWM mode, the GRB value represents the period value.
		Capture mode - In rising-edge clear mode, the register stores the counter value captured on the rising edge of the signal at port TnIO. (The opposite edge to that of GRA) - In falling-edge clear mode, the register stores the counter value captured on the falling edge of the signal at port TnIO. (The opposite edge to that of GRA)

9.2.6 TIMERn_CNT: timer/counter n count register

TIMERn_CNT is a 32-bit register. The count is incremented based on the specified input clock. This register can be both read and written to.

TIMER0_CNT=0x4000_3014, TIMER1_CNT=0x4000_3054, TIMER2_CNT=0x4000_3094
TIMER3_CNT=0x4000_30D4, TIMER4_CNT=0x4000_3114, TIMER5_CNT=0x4000_3154
TIMER6_CNT=0x4000_3194, TIMER7_CNT=0x4000_31D4

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	
-		RW	

15 0	CNT	Timer count value	
		R	Reads the current timer count.
		W	Sets the count value.

9.2.7 TIMERn_SR: timer/counter n status register

TIMERn_SR is an 8-bit register that shows the timer module’s current status.

TIMER0_SR=0x4000_3018, TIMER1_SR=0x4000_3058, TIMER2_SR=0x4000_3098
TIMER3_SR=0x4000_30D8, TIMER4_SR=0x4000_3118, TIMER5_SR=0x4000_3158
TIMER6_SR=0x4000_3198, TIMER7_SR=0x4000_31D8

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				MFA	MFB	OVF
-				0	0	0
-				RC	RC	RC

2	MFA	GRA match flag	
		0	No GRA match has been detected.
		1	A GRA match has been flagged. (Writing a 1 to the bit clears it.)
1	MFB	GRB match flag	
		0	No GRB match has been detected.
		1	A GRB match has been flagged. (Writing a 1 to the bit clears it.)
0	OVF	Counter overflow flag	
		0	No overflow event has occurred.
		1	A counter overflow event has been flagged. (Writing a 1 to the bit clears it.)

9.2.8 TIMERn_IER: timer/counter n interrupt enable register

TIMERn_IER register is 8-bits wide. Each status flag in the timer block can generate an interrupt. To trigger an interrupt, you must write a 1 to the corresponding bit in TIMERn_IER.

TIMER0_IER=0x4000_301C, TIMER1_IER=0x4000_305C, TIMER2_IER=0x4000_309C
TIMER3_IER=0x4000_30DC, TIMER4_IER=0x4000_311C, TIMER5_IER=0x4000_315C
TIMER6_IER=0x4000_319C, TIMER7_IER=0x4000_31DC

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																													MAIE	MBIE	OVIE
-																													0	0	0
-																													RW	RW	RW

2	MAIE	Whether to enable or disable the GRA match interrupt	
		0	Disables the GRA match interrupt.
		1	Enables the GRA match interrupt.
1	MBIE	Whether to enable or disable the GRB match interrupt	
		0	Disables the GRB match interrupt.
		1	Enables the GRB match interrupt.
0	OVIE	Whether to enable or disable the counter overflow interrupt	
		0	Disables the counter overflow interrupt.
		1	Enables the counter overflow interrupt.

9.2.9 TIMERn_TRGPNT: timer/counter n trigger point register

TIMERn_TRGPNT is a 32-bit register. The register is used to trigger the ADC at the desired timer counter point.

TIMER0_TRGPNT=0x4000_3020, TIMER1_TRGPNT=0x4000_3060, TIMER2_TRGPNT=0x4000_30A0
TIMER3_TRGPNT =0x4000_30E0, TIMER4_TRGPNT =0x4000_3120, TIMER5_TRGPNT =0x4000_3160
TIMER6_TRGPNT =0x4000_31A0, TIMER7_TRGPNT =0x4000_31E0

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																TRGPNT															
-																0x0000															
-																RW															

15 0	TRGPNT	Timer trigger point value	
		R	Reads the current trigger point value.
		W	Sets the trigger point value.

NOTE: The ADC-triggering value can be read when the timer count matches the set value.

9.2.10 TIMERn_SYNC: timer/counter n sync configuration register

TIMERn_SYNC is used to synchronize timers 0–3/4–7 to start or clear the respective timers. For example, if you want T1 to start running 0xF counts after the start of T0, you must synchronize T1 in the TIMER0_SYNC register and write 0xF to the delay bit field. Enabling T0SYNCB in the TIMER0_SYNC register, or synchronizing T1 with TIMER0_SYNC and then synchronizing T0 with TIMER1_SYNC will have no effect.

TIMER0_SYNC=0x4000_3024, TIMER1_SYNC=0x4000_3064, TIMER2_SYNC=0x4000_30A4
TIMER3_SYNC=0x4000_30E4, TIMER4_SYNC=0x4000_3124, TIMER5_SYNC=0x4000_3168
TIMER6_SYNC=0x4000_31A8, TIMER7_SYNC=0x4000_31E8

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				T7SYNCB	T6SYNCB	T5SYNCB	T4SYNCB	T3SYNCB	T2SYNCB	T1SYNCB	T0SYNCB	Reserved		SSYNC	CSYNC	SYNCDLY															
-				0	0	0	0	0	0	0	0	-		0	0	0x0000															
-				RW	RW	RW	RW	RW	RW	RW	RW	-		RW	RW	RW															

20+x	TxSYNCB (x=0–7)	Whether or not to synchronize the timer with Tx GROUP1: T0–T3, GROUP2: T4–T7	
		0	Disables synchronization.
		1	Enables synchronization.
NOTE: Synchronization is configurable within the same group. The timer to be master must select the slave timer to be synchronously activated, and the timer to be slave must not select the timer to be master.			
17	SSYNC	Whether or not to synchronize the start counter with another timer. (The slave timer also needs this configuration)	
		0	No synchronization
		1	Synchronized counter start mode
16	CSYNC	Whether or not to synchronize the clear counter with another timer. (The slave timer also needs this configuration)	
		0	No synchronization
		1	Synchronized counter clear mode
0	SYNCDLY	Start delay count value between synchronized timers (Applies to the master counter)	
		This bit field is used to set the interval between the starting points of two synchronized timers. If the bit field is set to zero, the timers will start simultaneously. (In this case, the timers must not be running. If a value other than zero is written while the timers are running, the interval is applied from the next cycle onward.)	

NOTES:

1. The synchronized slave timer starts counting when the master counter’s count value reaches the value written to SYNCDLY. And the synchronized timers are cleared simultaneously.
2. If you enable synchronization with T1 in TIMER0_SYNC and synchronization with T2 in TIMER1_SYNC, T0, T1 and T2 will start successively with the delay time set at SYNCDLY between them.

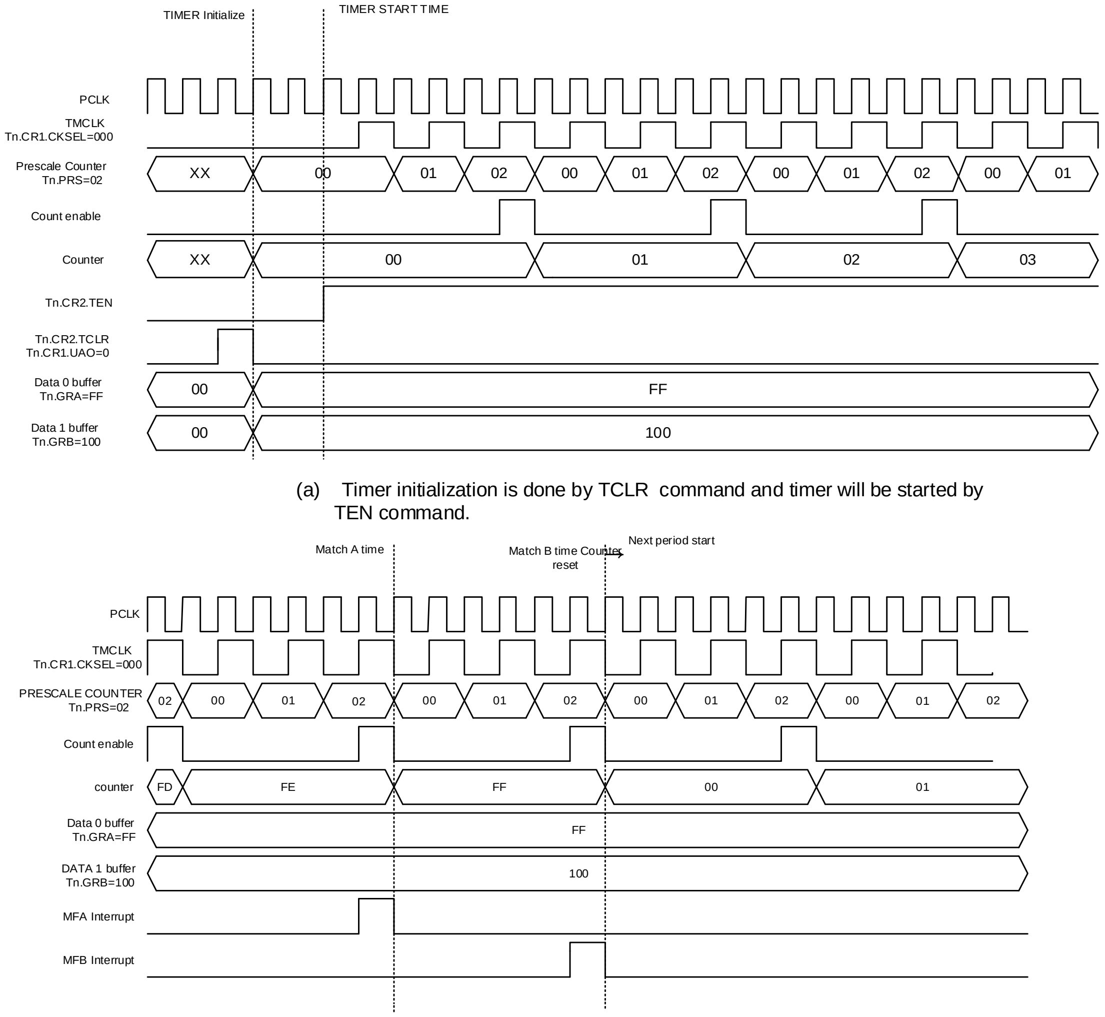
9.3 Functional description

9.3.1 Basic timer operations

The TMCLK shown in Figure 50 is a reference clock for operating the timer. The frequency of this clock can be divided by setting the prescaler to operate a counting clock.

The figure below shows the start and end points of a counter in normal periodic mode.

Figure 50. Basic Start and Match Operations



(a) Timer initialization is done by TCLR command and timer will be started by TEN command.

(b) Timer will be reset by match GRB timing and count again from 00.

The following formula calculates the timer’s count period:

Period = TMCLK period * GRB value

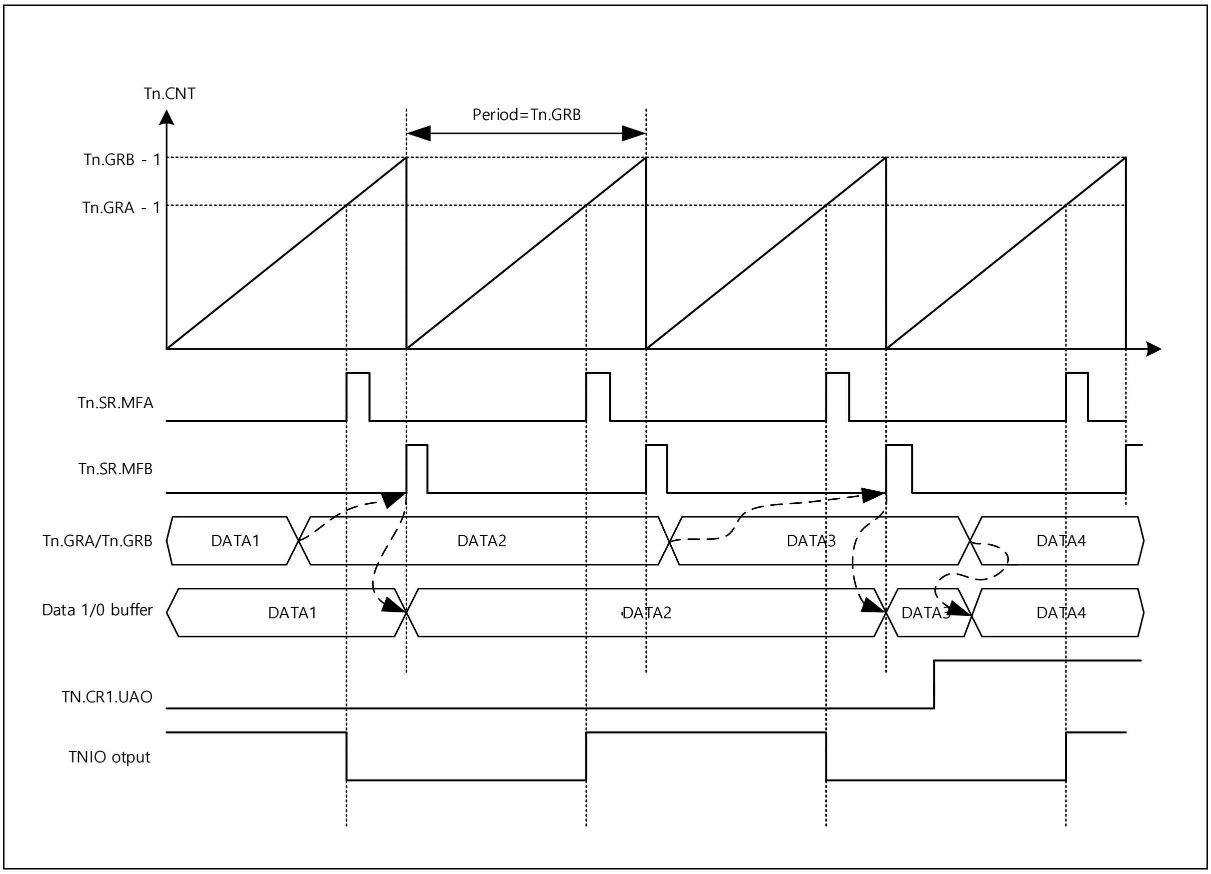
Match A interrupt time = TMCLK period * GRA value

When you change the timer setting or restart the timer with a new value, it is recommended that you write to CR2’s TCLR bit before writing to CR2’s TEN bit.

9.3.2 Normal periodic mode

Figure 51 shows the timing diagram for normal periodic mode.

Figure 51. Normal Periodic Mode Operation



The following formula calculates the timer’s count period:

Period = TMCLK period * GRB value

Match A interrupt time = TMCLK period * GRA value

If GRB = 0, the timer cannot be started even if CR2.TEN is 1 because the period is 0.

GRB value should not be 0. Also, the GRB value must be the same or greater than the GRA.

The values in GRA and GRB are loaded into the internal compare data buffers 0 and 1 when the loading condition occurs. In this periodic mode with CR1.UAO = 0, the CR2.TCLR write operation and the GRB match event will load the compare data buffers.

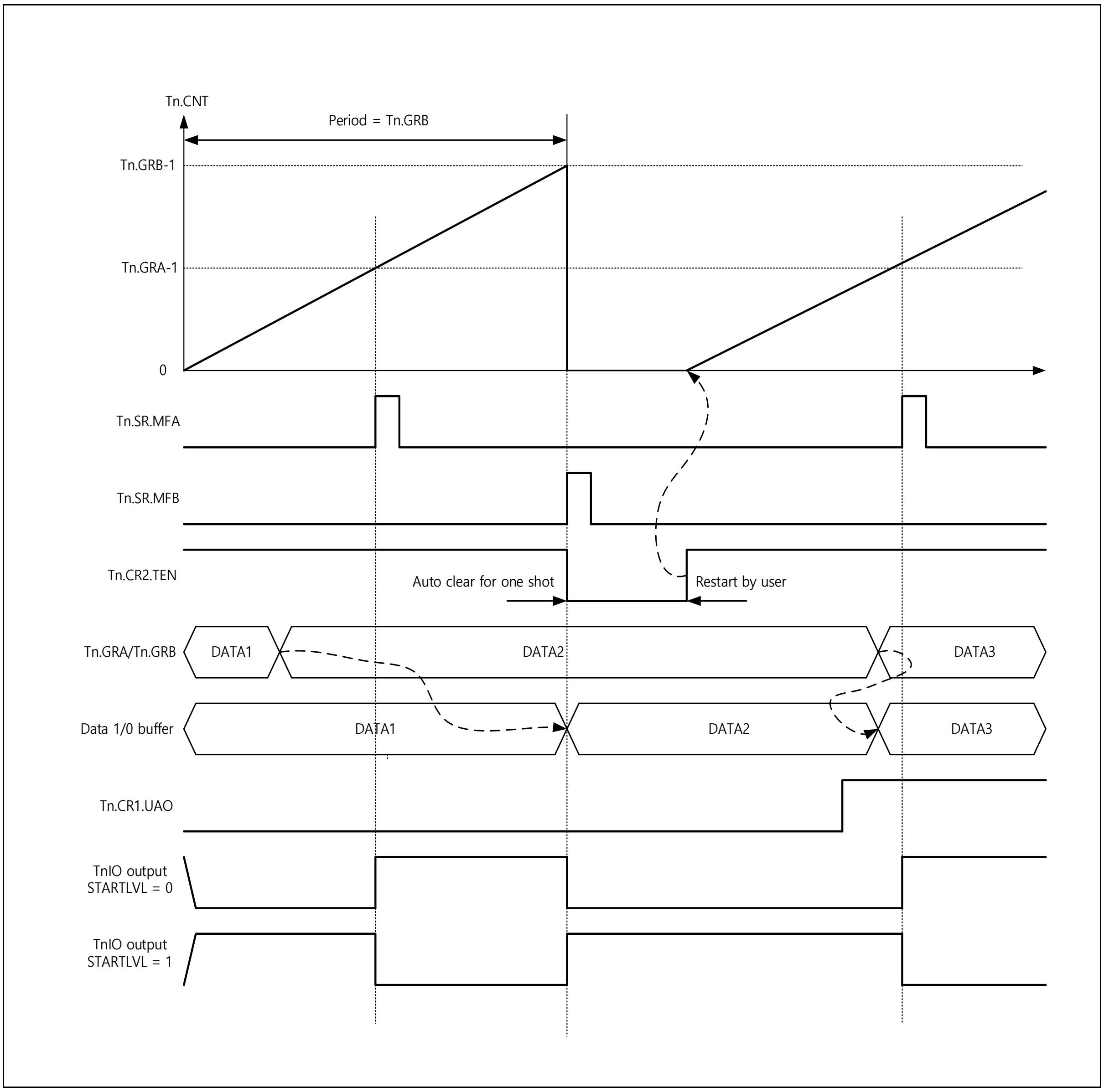
When CR1.UAO is 1, the internal compare data buffer is updated whenever the GRA or GRB data is updated.

The TnIO output signal will be toggled at every Match A condition time. If the value of GRA is 0, the TnIO output does not change its previous level. If GRA is the same as GRB, the TnIO output will toggle at same time as the counter start time. The initial level of the TnIO signal is decided by the CR1.STARTLVL value.

9.3.3 One-shot mode

Figure 52 shows the timing diagram for One-shot mode. In the figure, the GRB value determines the One-shot period, and the GRA value provides another comparative point.

Figure 52. One-shot Mode Operation



The following formula calculates the One-shot count period:

Period = TMCLK period * GRB value

Match A interrupt time = TMCLK period * GRA value

If GRB = 0, the timer cannot be started even if CR2.TEN is 1 because the period is 0.

The values in GRA and GRB are loaded into the internal compare data buffers 0 and 1 when the loading condition is met. In this periodic mode with CR1.UAO =0, the CR2.TCLR write operation and the GRB match event will load the compare data buffers.

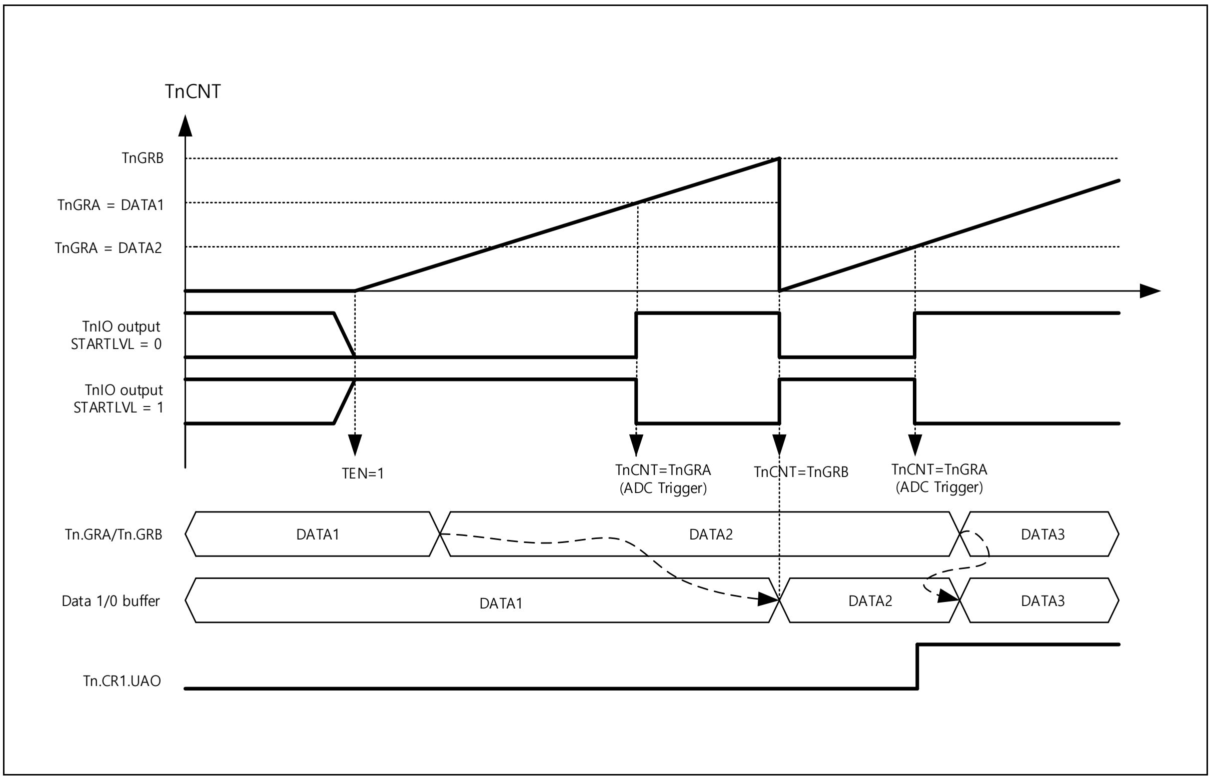
When CR1.UAO is 1, the internal compare data buffer is updated whenever the GRA or GRB data is updated.

The TnIO output signal format is the same as PWM mode. The GRB value defines the output pulse period and the GRA value defines the pulse width of one shot pulse.

9.3.4 PWM timer output

Figure 53 shows the timing diagram in PWM output mode. The TnGRB value decides the PWM pulse period. An additional comparison point is provided by the TnGRA register value which defines the pulse width of PWM output.

Figure 53. PWM Output Operation



The following formula calculates the PWM pulse period:

Period = TMCLK period * TnGRB value

Match A interrupt time = TMCLK period * TnGRA value

If GRB = 0, the timer cannot be started even CR2.TEN is 1 because the period is 0.

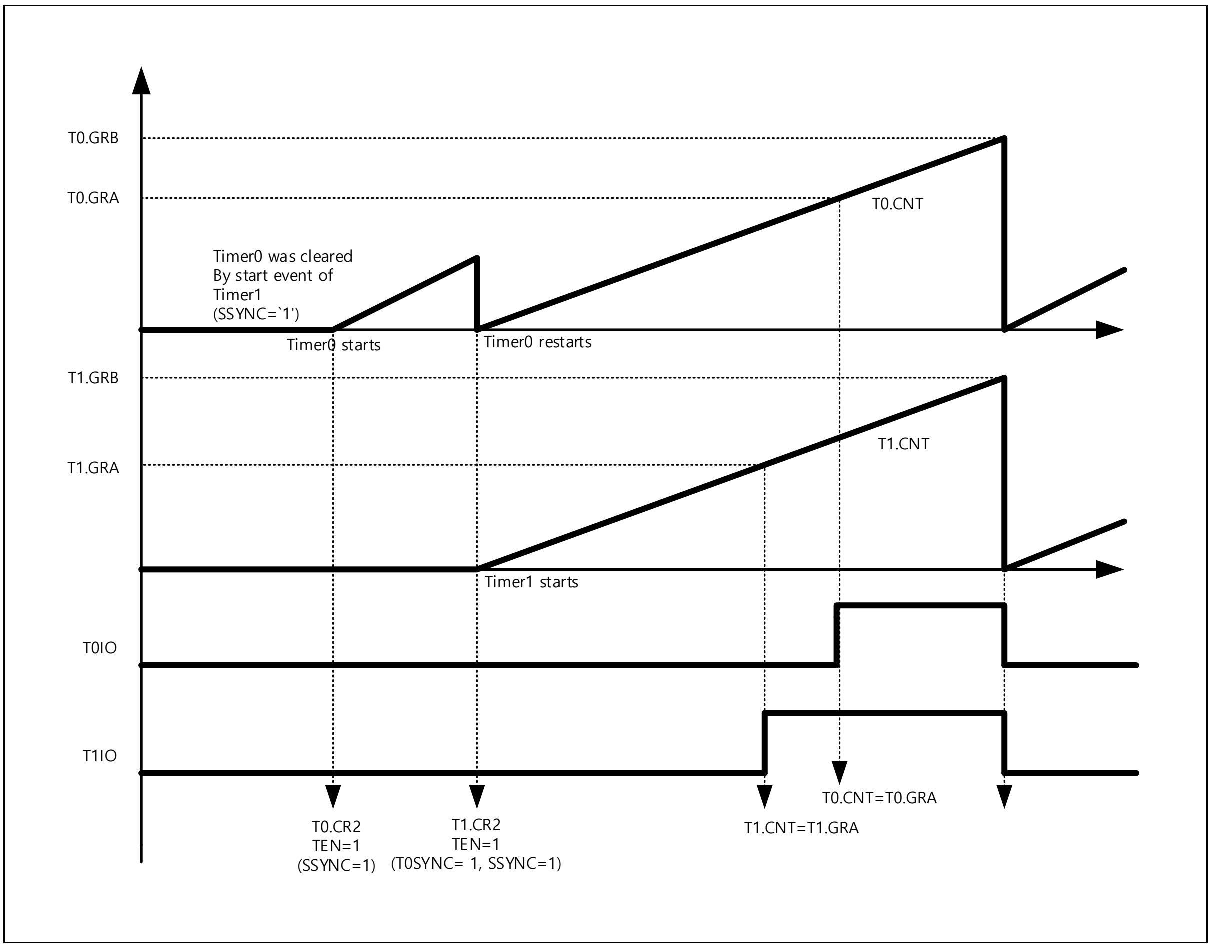
The values in GRA and GRB are loaded into the internal compare data buffers 0 and 1 when the loading condition is met. In this periodic mode with CR1.UAO =0, the CR2.TCLR write operation and the GRB match event will load the compare data buffers. When CR1.UAO is 1, the internal compare data buffer is updated whenever the GRA or GRB data is updated.

The TnIO output signal generates a PWM pulse. The GRB value defines the output pulse period and the GRA value defines the pulse width of one shot pulse. The active level of the PWM pulse can be controlled by the CR1.STARTLVL bit value. ADC trigger generation is available at Match A interrupt time.

9.3.5 PWM synchronization

2-PWM outputs are usually used as synchronous PWM signal control. This function is provided with synchronous start. Figure 54 shows synchronous PWM generation.

Figure 54. Timer Synchronization Example: When SSYNC = 1



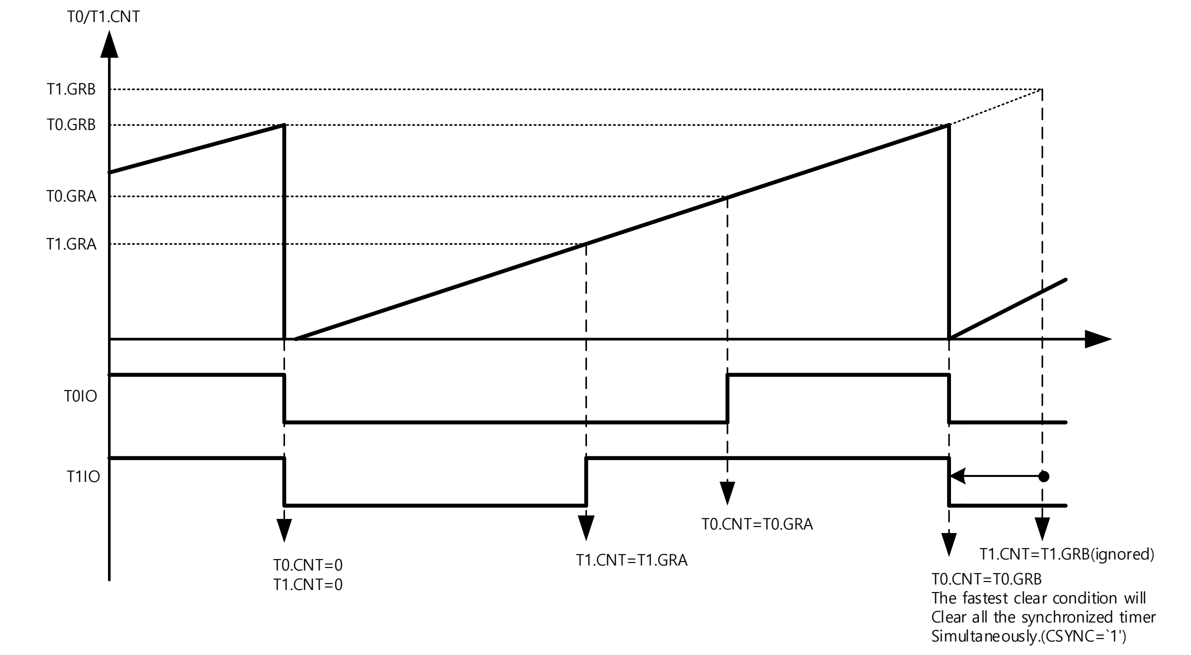
The TIMER1n_SYNC.SSYNC bit controls start synchronization with other timer blocks. In the above figure, the Timer 0 (slave timer) waveform shows that the timer's start is synchronized to the start of the signal from Timer 1 (master timer).

For this SYNC function as shown in the figure above, both the master and slave timers' synchronous start bits must be set enabled: In the master timer's TIMER1_SYNC register, T0SYNCB = 1 and SSYNC = 1 must be set. And in the slave timer's TIMER0_SYNC register, SSYNC = 1 must be set.

If only the master timer's synchronous start bit is set enabled for the slave timer while the slave's synchronous start bit is disabled, the slave timer can run independently without being affected by the master timer's start synchronization signal.

Additionally, because Timer 0 (slave) operates as synchronized to Timer 1's (master's) start synchronization signal, it runs even if the TIMER0_CR2.TEN bit is set to 0.

Figure 55. Timer Synchronization Example: When CSYNC = 1



The TIMER1n_SYNC.CSYNC bit controls clear synchronization with other timer blocks. In the above figure, the Timer 1 (slave timer) waveform shows that the timer's clear is synchronized to the clear of the signal from Timer 0 (master timer).

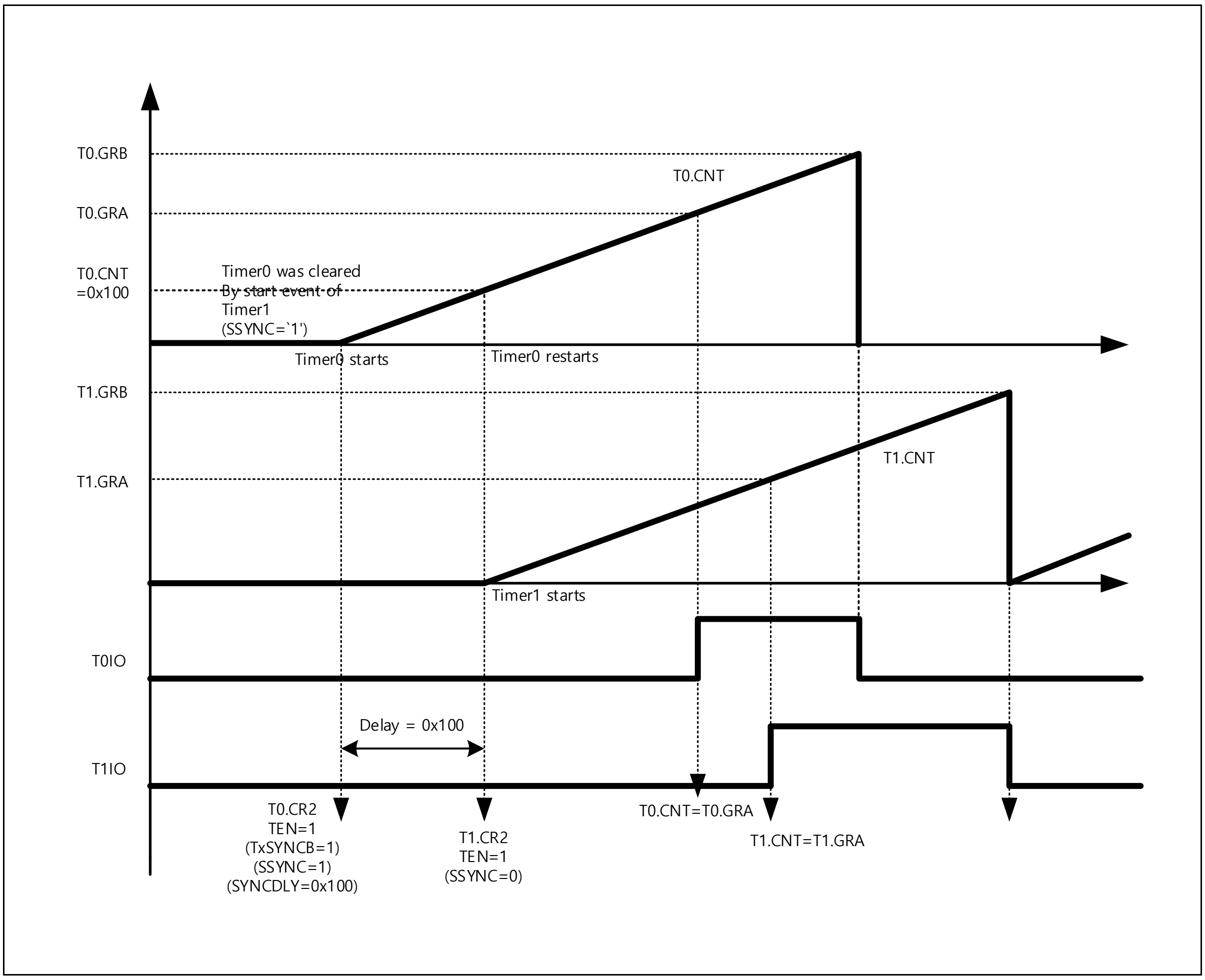
For this CSYNC function as shown in the figure above, both the master and slave timers' synchronous clear bits must be set enabled: In the master timer's TIMER0_SYNC register, T1SYNCB = 1 and CSYNC = 1 must be set. And in the slave timer's TIMER1_SYNC register, CSYNC = 1 must be set.

If only the master timer's synchronous clear bit is set enabled for the slave timer while the slave's synchronous clear bit is disabled, the slave timer can run independently without being affected by the master timer's clear synchronization signal.

9.3.6 PWM delayed synchronization

The PWM delayed synchronization function is used between timers within the same group (GROUP1: T0–T3, GROUP2: T4–T7); synchronization is delayed by the amount of time written to the TIMERn_SYNC register's SYNCDLY[15:0] bit. Using this function, you can have slave timers start sequentially after a certain amount of delay time has passed since the start of the master timer.

Figure 56. Timer Delayed Synchronization Example: When SSYNC = 1



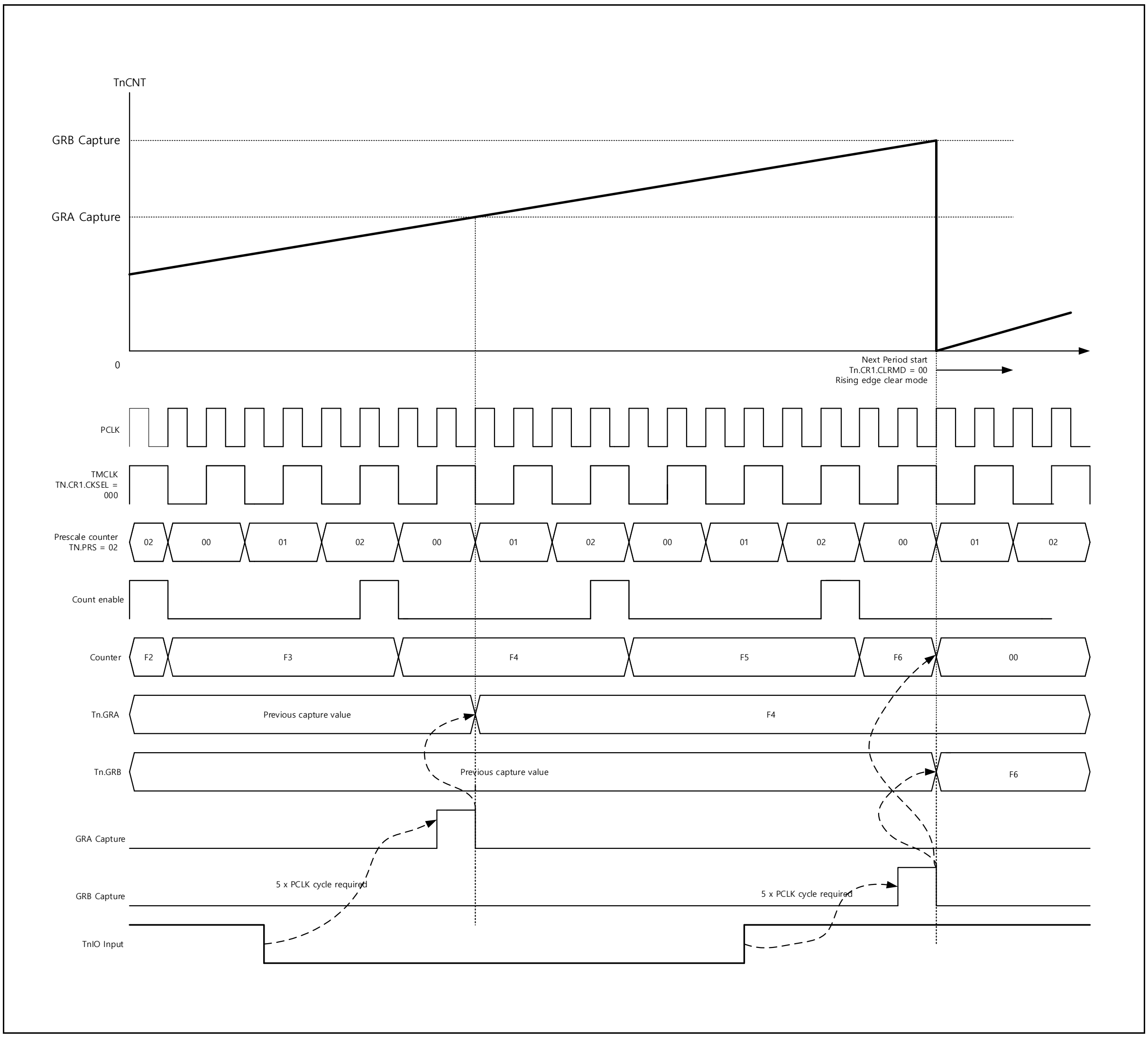
For example, to start Timer 1 (slave) after 0x100 counts from Timer 0's (master's) synchronization signal output as illustrated above, you must set the master timer's TIMER0_SYNC register as T1SYNCB = 1, SSYNC = 1, and SYNCDLY = 0x100. And then, you must set the slave timer's TIMER1_SYNC register as SSYNC = 1.

Additionally, because Timer 1 (slave) operates as synchronized to Timer 0's (master's) start signal, it runs even if the TIMERn_CR2.TEN bit is set to 0.

9.3.7 Capture mode

Figure 57 shows the timing diagram in the capture mode operation. The TnIO input signal is used for capturing the pulse. Rising and falling edges can capture the counter value in each capture condition.

Figure 57. Capture Mode Operation



A PCLK clock cycle is required internally. Therefore, the actual capture point is five PCLK clock cycles later from the rising or falling edge of the TnIO input signal.

The internal counter can be cleared in multiple modes. The TIMERn_CR1.CLRMD bit field controls counter clearing in capture mode. The supported modes include rising-edge clear mode, falling-edge clear mode, both-edge clear mode, and non-clear mode.

The example in Figure 57 is of rising-edge clear mode. On the falling edge of the input signal to TnIO, the TIMERn_GRA register captures the CNT value; on the rising edge, the TIMERn_GRB register captures the CNT value.

9.3.8 ADC triggering

The timer module can generate ADC start trigger signals. One timer can be one trigger source of the ADC block. Trigger source control is performed by the ADC control register.

The figures below illustrate how ADC triggering works in each mode. The conversion rate must be shorter than the timer period; otherwise, an overrun can occur. ADC acknowledge is not required because the trigger signal is automatically cleared three PCLK clock pulses later.

Figure 58. ADC Triggering Timing Diagram: TRGMOD=0x00, 0x01

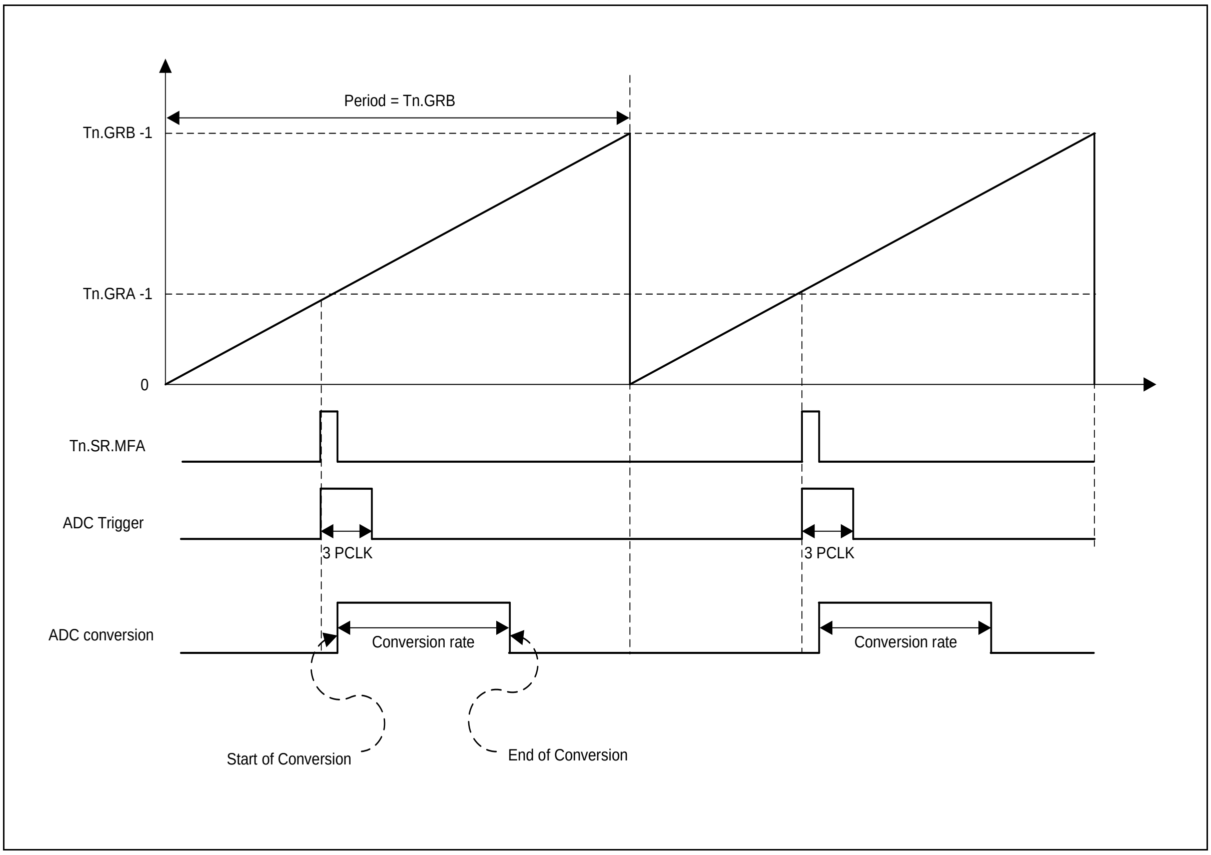


Figure 59. ADC Triggering Timing Diagram: TRGMOD=0x10

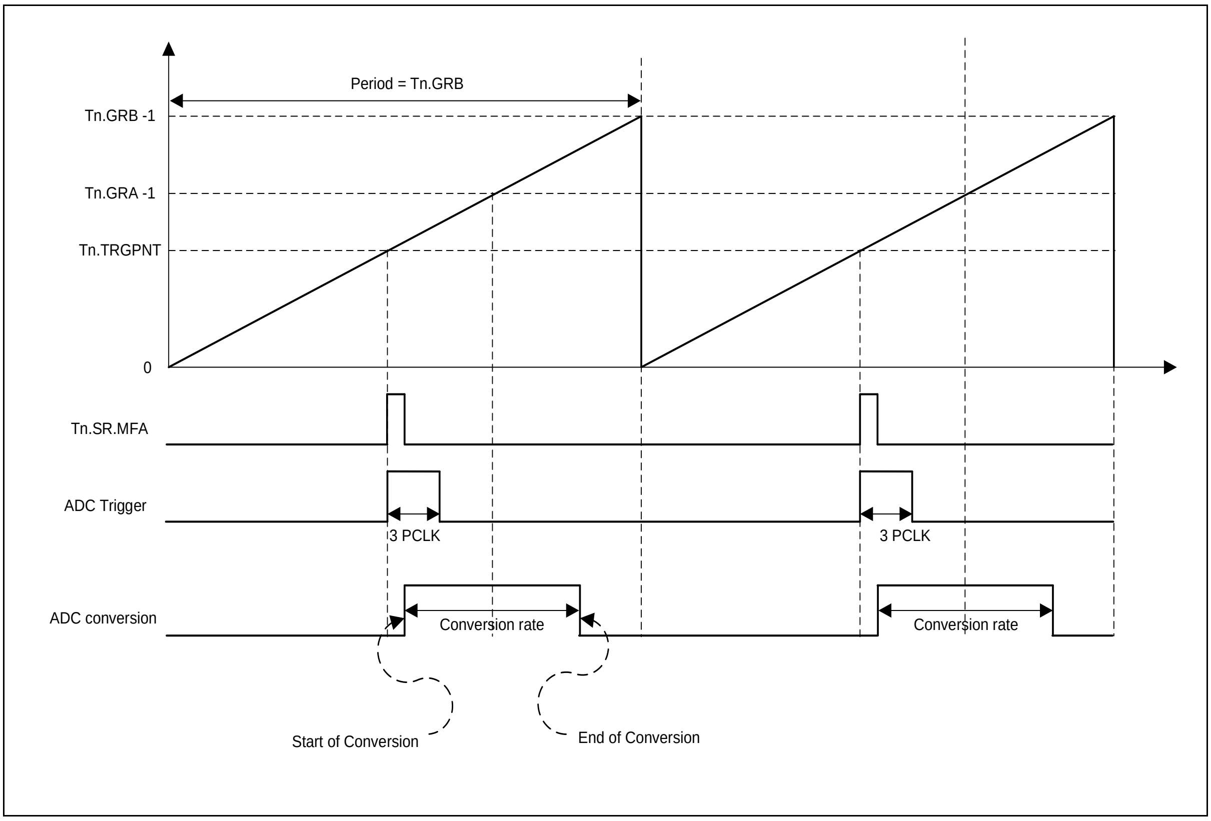
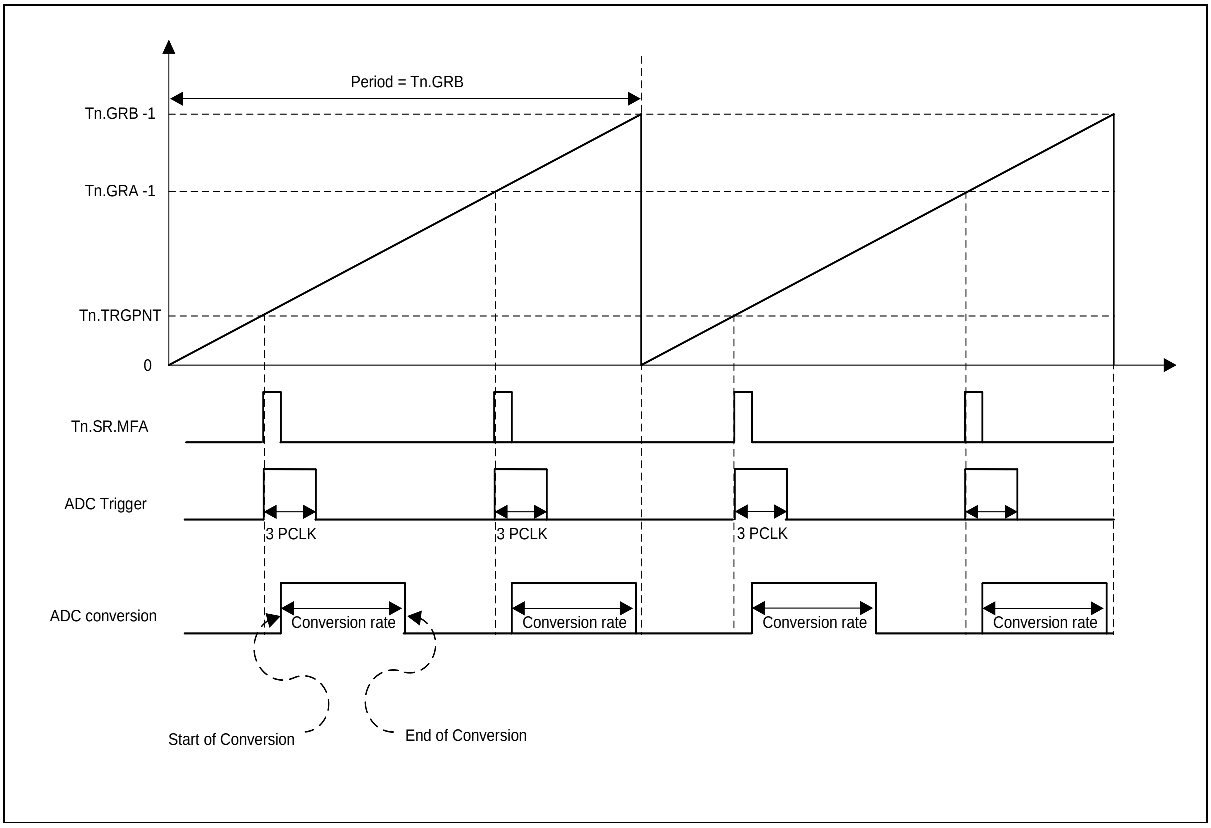


Figure 60. ADC Triggering Timing Diagram: TRGMOD=0x11



9.3.9 Setting examples

<Example 1> Timer periodic mode (PCLK = 8MHz, period = 1 ms)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                    : Unlocks the SCU registers.
SCU_PER1<TIMER0[16]> = “1”                          : Enables the TIMER0 peripheral.
SCU_PCER1<TIMER0[16]> = “1”                         : Enables the TIMER0 peripheral clock.

TIMER0_CR2<TCLR[1]> = “1”                           : Initializes the timer.
TIMER0_CR2<TCLR[1]> = “0”
TIMER0_CR1<MODE[1:0]> = “00”                        : Sets the timer mode to periodic mode.
TIMER0_CR1<CKSEL[6:4]> = “000”                      : Selects the timer clock source (8MHz/2) = 250ns.
TIMER0_PRS<PRS[9:0]> = “0x00”                       : Specifies the frequency divider value for the timer clock.

TIMER0_GRA<GRA[15:0]> = “00001111_10100000”         : Sets the GRA value to 4000 (250ns * 4000) = 1ms.
TIMER0_GRB<GRB[15:0]> = “00000000_00000000”         : Initializes the GRB register.
TIMER0_CNT<CNT[15:0]> = “00000000_00000000”         : Initializes the timer counter.

TIMER0_SR<MFA[2]> = “0”                             : Clears the timer’s GRA match interrupt.
TIMER0_IER<MAIE[2]> = “1”                           : Enables the GRA match interrupt 0 in timer   periodic mode.
TIMER0_CR2<TEN[0]> = “1”                            : Enables the timer.
```

10 Free Run Timer (FRT)

A33M11x series has one Free Run Timer (FRT), which is a built-in 32-bit up-count timer. This timer can run with the overflow or a match interrupt according to its uses and can remain active in stop mode.

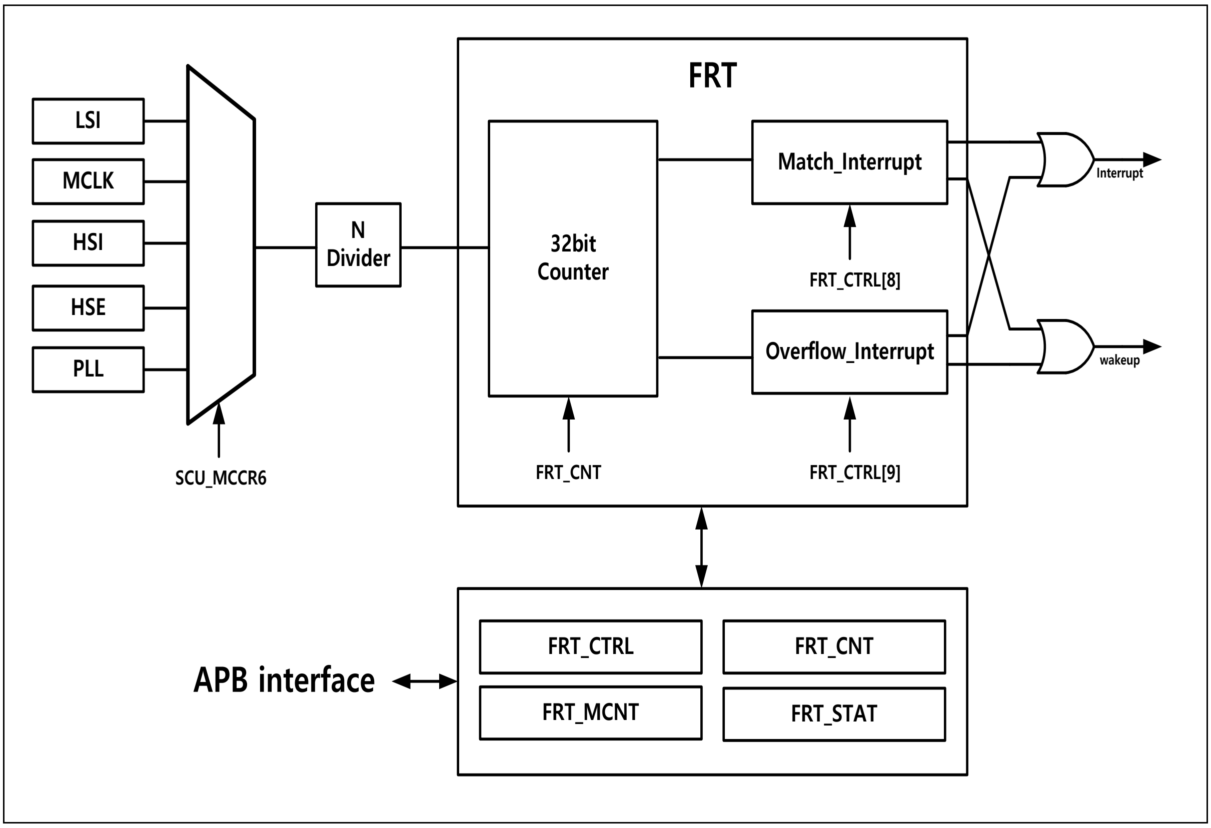
The FRT of A33M11x series features the followings:

- 32-bit up-count timers
  - Capable of functioning as periodic timers (Each timer's period is configurable)
  - Free-run timer mode
- FRT overflow and match interrupts supported
- FRT input clock sources selectable
  - Clock sources selectable with the setting of SCU_MCCR6: LSI, MCLK, HSI, HSE, and PLL

10.1 FRT block diagram

In this section, FRT block diagram is introduced in Figure 61.

Figure 61. FRT Block Diagram



10.2 Registers

Base address of FRT block is introduced in the followings:

Table 42. Base Address of FRT Interface

Name	Base address
FRT	0x4000_0600

Table 43. FRT Register Map

Name	Offset	Type	Description	Reset value	Reference
FRT_CTRL	0x0000	RW	FRT control register	0x0000_0000	10.2.1
FRT_MCNT	0x0004	RW	FRT match counter register	0x0000_0000	10.2.2
FRT_CNT	0x0008	RW	FRT counter register	0x0000_0000	10.2.3
FRT_STAT	0x000C	RC	FRT status register	0x0000_0000	10.2.4

10.2.1 FRT_CTRL: FRT control register

FRT_CTRL is a 32-bit register. Its bits control the operation of the FRT.

FRT_CTRL=0x4000_0600

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	OVFIE	MATCHIE	Reserved	MODE	EN
-	0	0	-	0	0
-	RW	RW	-	RW	RW

9	OVFIE	Whether to enable or disable the FRT counter overflow interrupt	
		0	Disables the overflow interrupt.
		1	Enables the overflow interrupt.
8	MATCHIE	Whether to enable or disable the FRT counter match interrupt	
		0	Disables the match interrupt.
		1	Enables the match interrupt.
1	MODE	FRT mode selection	
		0	Free run timer mode
		1	Match interrupt mode
0	EN	Whether to enable or disable the FRT	
		0	Disables.
		1	Enables.

10.2.2 FRT_MCNT: FRT match counter register

FRT_MCNT is a 32-bit register. It is used to specify the period value when the FRT operates in periodic timer mode.

In match interrupt mode, the value of the FRT_CNT register counts up until it reaches the FRT_MCNT value, which triggers the match interrupt if it has been set enabled.

FRT_MCNT=0x4000_0604

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
MCNT																															
0x0000_0000																															
RW																															

31 0	MCNT	FRT’s match counter value
		The match interrupt is triggered when the current counter value reaches the set match counter value.

10.2.3 FRT_CNT: FRT counter register

FRT_CNT is a 32-bit register that shows the timer’s current count value. The register can be both read and written to and functions as an up-count timer, whose count value is incremented.

If FRT_CTRL’s EN bit is set to 1, FRT_CNT can be read and written to.

FRT_CNT=0x4000_0608

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																															
0x0000_0000																															
RW																															

31 0	CNT	FRT count data
		Represents the current count value. (Only the initializing value 0x0 can be written to the bit field.)

10.2.4 FRT_STAT: FRT status register

FRT_STAT is a 32-bit register that shows the FRT status.

FRT_STAT=0x4000_060C

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	OVFI	MATCHI	Reserved
-	0	0	-
-	RW	RW	-

9	OVFI	FRT counter overflow interrupt status	
		0	The overflow event has not occurred.
		1	The overflow event has occurred. (writing a 1 to the bit clears the flag)
8	MATCHI	FRT counter match interrupt status	
		0	The match interrupt event has not occurred.
		1	The match interrupt event has occurred. (writing a 1 to the bit clears the flag)

10.3 Functional description

The FRT_CTRL register’s MODE bit determines whether the FRT will operate in free-run mode or periodic timer mode.

When it is run in free-run mode, the FRT_CTRL register’s OVFIE bit determines whether to enable or disable the interrupt. The MATCHIE bit does not affect the interrupt.

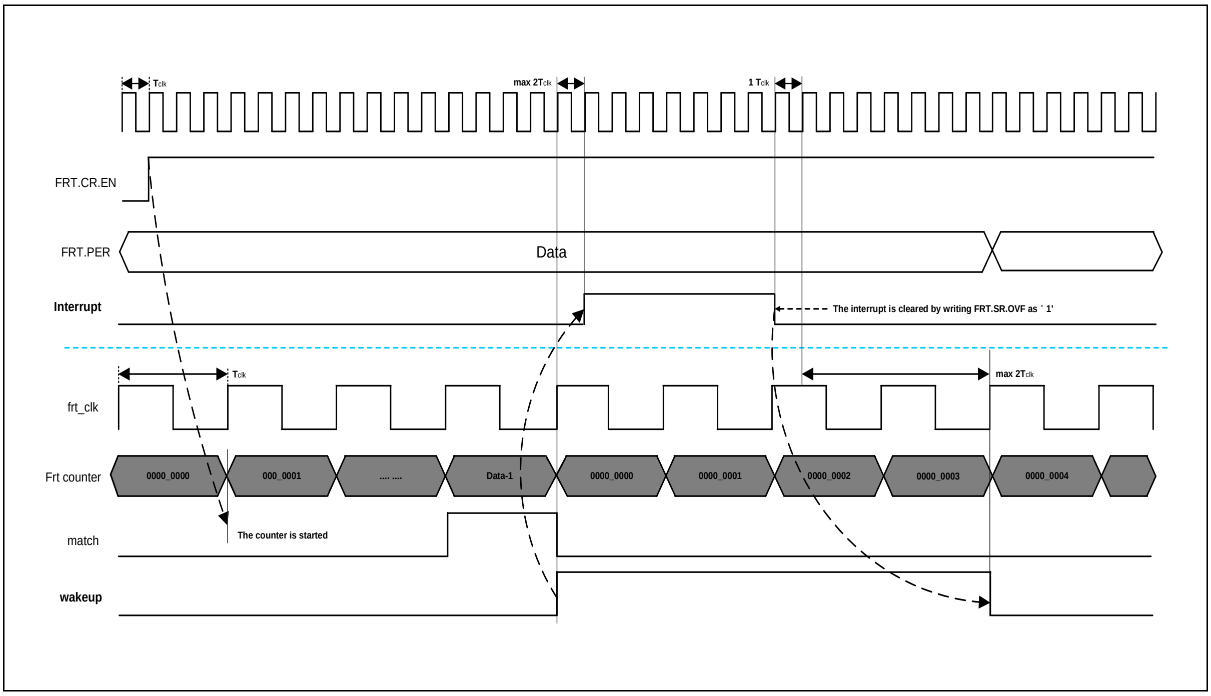
When it is run in periodic timer mode, the FRT_CTRL register’s MATCHIE bit determines whether to enable or disable the interrupt. The OVFIE bit does not affect the interrupt.

10.3.1 Match interrupt operation

Figure 62 is a diagram that illustrates match interrupt operation. To enable the match interrupt, FRT_CTRL’s MATCHIE bit must be set to 1.

When FTR_CTRL’s EN bit is set to 1, the FRT counter starts counting. Once the counter value reaches the FRT_MCNT value, the interrupt and wake-up signal are triggered. An interrupt signal can be delayed up to approximately two system clock pulses. A wake-up signal can be delayed up to approximately 1 clk + 2 frt_clk.

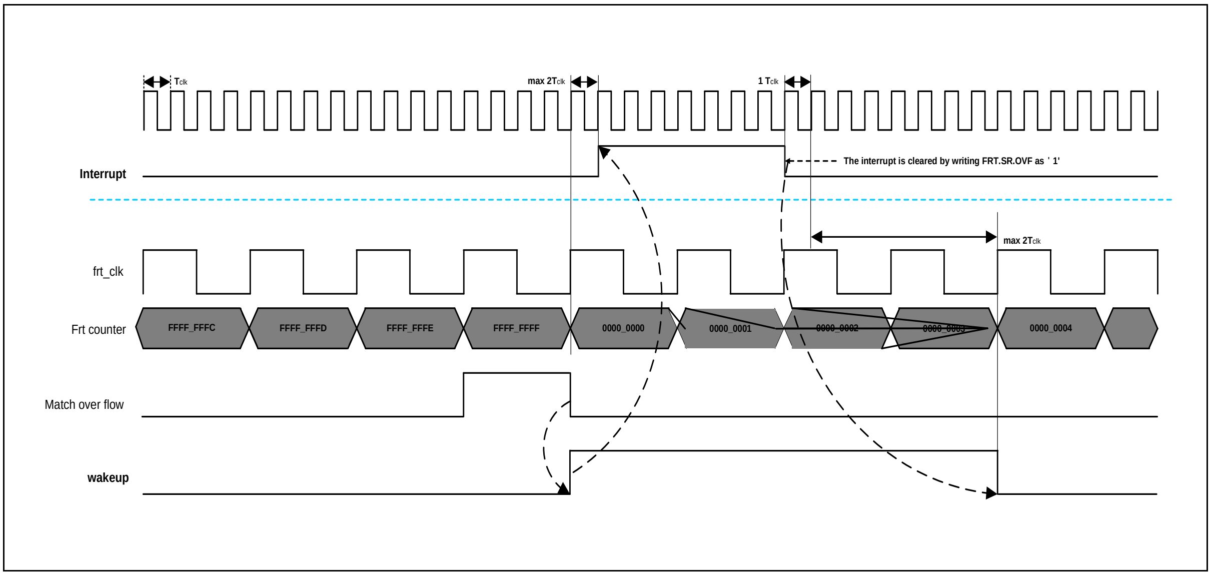
Figure 62. Match Interrupt Operation Timing Diagram



10.3.2 Overflow interrupt operation

Figure 63 shows a diagram of overflow interrupt timing. The overflow interrupt operates almost in the same way as the match interrupt. The overflow interrupt is triggered when the FRT counter value matches 0xFFFFFFFF.

Figure 63. Overflow Interrupt Operation Timing Diagram



10.3.3 Setting examples

<Example 1> Setting free-run mode (8MHz HSE, 0–0xFFFFFFFF)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                   : Unlocks the SCU registers.
SCU_PER1<FRT[6]> = “1”                             : Enables the FRT peripheral.
SCU_PCER1<FRT[6]> = “1”                            : Enables the FRT peripheral clock.

SCU_MCCR6<FRT0CDIV[10:8]> = “110”                  : Selects the 8MHz HSE as the FRT clock source.

FRT_CTRL<MODE[1]> = “0”                            : Sets FRT mode to free-run mode.
FRT_CTRL<OVFIE[9]> = “1”                           : Enables the FRT overflow event.

NVICIP[8]<PRI_8[7:0]> = “00110000”                 : Sets the NVIC FRT interrupt’s priority level.
NVICISER[0]<SETPEND[31:0]>
= “00000000_00000000_00000000_00010000”

FRT_CTRL<EN[0]> = “1”                              : Enables the NVIC FRT interrupt.
FRT_CNT<CNT[31:0]>                                 : Enables the FRT counter.
=”00000000_00000000_00000000_00000000”             : Initializes the FRT counter value to zero.
```

11 Universal Asynchronous Receiver/Transmitter (UART)

A33M11x series is equipped with a four-channel UART module. These built-in UARTs transmit and receive data according to user-specified settings and read the current UART status. UART status information includes the type and conditions of the current UART transmission/reception process and can be used to check for errors (parity, overrun, framing, or break interrupts) that occur during data reception.

Each UART channel has a programmable baud-rate generator, which serves to generate an internal clock for the corresponding UART by dividing the prescaled clock by a baud-rate divisor (ranging from 1 to 65535) and then dividing the result by 16.

Additionally, the user can program interrupts that control UART communication.

UART of A33M11x series features the followings:

- A total of four 16450 asynchronous serial communication ports supported
- Configurable standard asynchronous communication bits (start, stop, and parity)
- User-programmable serial communication
  - 5, 6, 7, or 8 data bits
  - Even, odd, or no parity generation and checking
  - 1-, 1.5-, or 2-stop bit generation and checking
- A 16-bit baud-rate generator and an 8-bit fractional compensator
- Delay between data frames supported
- Transfer status indicated by the interrupt ID and line status registers
  - Stop bit error detection
  - Display of information about the current status
  - Line break generation and checking
  - Receive error diagnosis
- A priority-based interrupt system

Table 44 introduces pins assigned for the UART.

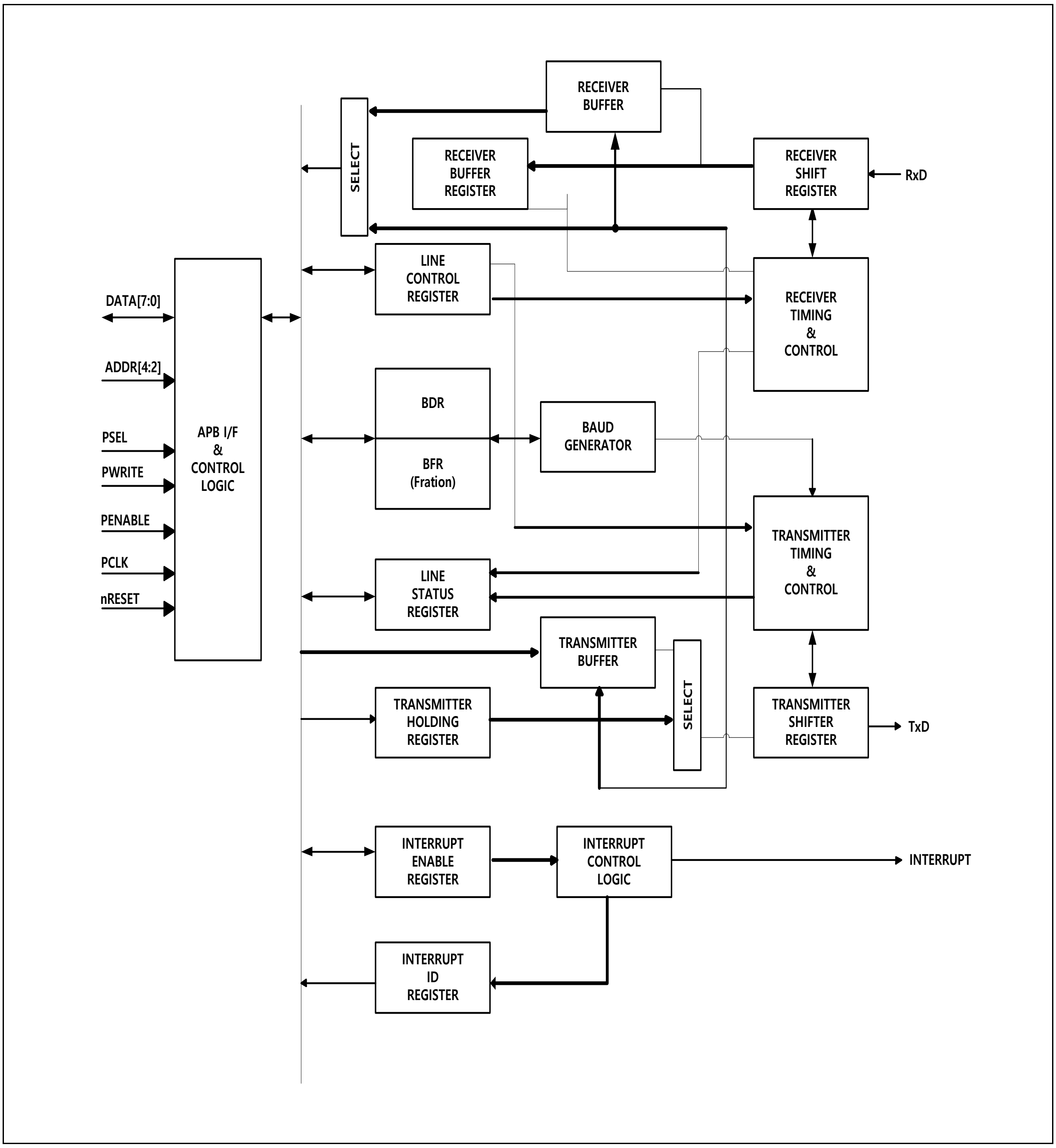
Table 44. Pin Assignment of UART: External Pins

Pin name	Type	Description	Supported Packages	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
TXD0	O	UART channel 0 transmit output	O	O
RXD0	I	UART channel 0 receive input	O	O
TXD1	O	UART channel 1 transmit output	O	X
RXD1	I	UART channel 1 receive input	O	X
TXD2	O	UART channel 2 transmit output	O	O
RXD2	I	UART channel 2 receive input	O	O
TXD3	O	UART channel 3 transmit output	O	O
RXD3	I	UART channel 3 receive input	O	O

11.1 UART block diagram

In this section, UART is introduced in block diagrams.

Figure 64. UART Block Diagram



11.2 Registers

Base address of UART is introduced in the followings:

Table 45. Base Address of UART

Name	Base address
UART0	0x4000_8000
UART1	0x4000_8100
UART2	0x4000_8200
UART3	0x4000_8300

Table 46. UART Register Map

Name	Offset	Type	Description	Reset value	Reference
UARTn_RBR	0x0000	RO	UART n receive data buffer register	0x0000_0000	11.2.1
UARTn_THR	0x0000	WO	UART n transmit data hold register	0x0000_0000	12.2.2
UARTn_IER	0x0004	RW	UART n interrupt enable register	0x0000_0000	11.2.3
UARTn_IIR	0x0008	RO	UART n interrupt ID register	0x0000_0001	11.2.4
UARTn_LCR	0x000C	RW	UART n line control register	0x0000_0003	11.2.5
UARTn_DCR	0x0010	RW	UART n data control register	0x0000_0000	11.2.6
UARTn_LSR	0x0014	RO	UART n line status register	0x0000_0060	11.2.7
UARTn_BDR	0x0020	RW	UART n baud-rate divisor latch register	0x0000_0000	11.2.8
UARTn_BFR	0x0024	RW	UART n baud-rate fraction register	0x0000_0000	11.2.9
UARTn_IDTR	0x0030	RW	UART n inter-frame delay time register	0x0000_0000	11.2.10

NOTE: n = 0, 1, 2 and 3

11.2.1 UARTn_RBR: UART n receive data buffer register

UARTn_RBR is an 8-bit read-only register. Received data is read from this register, and the maximum length of data is 8 bits. The last received data is retained until a new byte is received.

UART0_RBR=0x4000_8000, UART1_RBR=0x4000_8100, UART2_RBR=0x4000_8200
UART3_RBR=0x4000_8300

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			RBR[7:0]
-			-
-			RO

7 0	RBR	Receive buffer register

11.2.2 UARTn_THR: UART n transmit data hold register

UARTn_THR is an 8-bit write-only register. Data is stored in this register to be transmitted. However, data written to THR cannot be read but sent to the transmit shift register when this register is empty.

UART0_THR=0x4000_8000, UART1_THR=0x4000_8100, UART2_THR=0x4000_8200
UART3_THR=0x4000_8300

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			THR[7:0]
-			-
-			WO

7 0	THR	Transmit buffer register

11.2.3 UARTn_IER: UART n interrupt enable register

UARTn_IER enables six types of UART interrupts. Each interrupt generates an interrupt output signal. If you set all of the IER bits to 0, all UART interrupts become disabled. You can enable a particular interrupt by setting the corresponding bit to 1.

UART0_IER=0x4000_8004, UART1_IER=0x4000_8104, UART2_IER=0x4000_8204
UART3_IER=0x4000_8304

<table>
<tr><td colspan="8">31 30 29 28 27 26 25 24</td><td colspan="8">23 22 21 20 19 18 17 16</td><td colspan="8">15 14 13 12 11 10 9 8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="26">Reserved</td><td>DTXIEN</td><td>DRXIEN</td><td>TXEIE</td><td>RLSIE</td><td>THREIE</td><td>DRIE</td></tr>
<tr><td colspan="26">-</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="26">-</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td></tr>
</table>

5	DTXIEN	Whether to enable or disable the DMA transmit complete interrupt	
		0	Disables the DMA transmit complete interrupt.
		1	Enables the DMA transmit complete interrupt.
4	DRXIEN	Whether to enable or disable the DMA receive complete interrupt	
		0	Disables the DMA receive complete interrupt.
		1	Enables the DMA receive complete interrupt.
3	TXEIE	Whether to enable or disable the transmit complete interrupt	
		0	Disables the transmit complete interrupt.
		1	Enables the transmit complete interrupt.
2	RLSIE	Whether to enable or disable the receive line status interrupt	
		0	Disables the receive line status interrupt.
		1	Enables the receive line status interrupt.
1	THREIE	Whether to enable or disable the transmit data hold register empty (THRE) interrupt	
		0	Disables the THRE interrupt.
		1	Enables the THRE interrupt.
0	DRIE	Whether to enable or disable the data receive interrupt	
		0	Disables the data receive interrupt.
		1	Enables the data receive interrupt.

11.2.4 UARTn_IIR: UART n interrupt ID register

UARTn_IIR is a read-only register that informs of the occurrence of the interrupts set enabled in UARTn_IER.

When the CPU accesses UARTn_IIR, the UART locks all interrupts and the highest-priority interrupt is read. Interrupts occur even while the register is being accessed by the CPU; however, its status remains unchanged until the current access is completed.

UART0_IIR=0x4000_8008, UART1_IIR=0x4000_8108, UART2_IIR=0x4000_8208
UART3_IIR=0x4000_8308

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																											TXE	IID			IPEN
-																											0	000			1
-																											RO	RO			RO

4	TXE	Transmit complete (Refer to Table 47 for interrupt source IDs.)	
TXE Indicates whether or not the transmit data hold register (THR) is empty. TXE = 0 indicates that THR is currently empty and thus new data can be written to the register.			
3 1	IID	Interrupt source ID (Refer to Table 47 for interrupt source IDs.)	
IID Displays the interrupt with the highest priority among those currently active. Refer to Table 47 Interrupt IDs and Control for details.			
0	IPEN	Presence of pending interrupts	
		0	An interrupt is pending
		1	There are no interrupts pending
IPEN Indicates whether or not there are currently unprocessed interrupts. IPEN = 0 indicates that a certain interrupt condition has occurred, and IPEN = 1 means that there are no currently unprocessed interrupts.			

Among the pending interrupts, the highest-priority interrupt's source ID is indicated by the IID bit. The UART module uses a total of seven interrupts with different priorities. These interrupts include:

- Receive line status interrupt
- Receive data ready interrupt/character timeout interrupt
- Transmit data hold register empty (THRE) interrupt
- DMA Tx/Rx complete interrupts

Table 47. Interrupt IDs and Control

Priority level	TXE	DMA	IID		IPEN	Interrupt Source		
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Interrupt name	Interrupt condition	Interrupt clear
-	0	0	0	0	1	N/A	-	-
1	0	0	1	1	0	Receive line status	Overrun, parity, framing, break error, etc.	Read LSR
2	0	0	1	0	0	Receive data present	There is data received	Read the receive register or IIR
3	0	0	0	1	0	Transmit data hold register empty	The transmit buffer is empty	Write to the transmit data hold register or IIR
4	1	X	X	X	X	Transmit register empty	The transmit data hold register is empty	Write to the transmit data hold register or read IIR
5	0	1	1	0	0	DMA Rx complete	DMA Rx complete	Read IIR
6	0	1	0	1	0	DMA Tx complete	DMA Tx complete	Read IIR
7	1	1	X	X	X	Transmit register empty and DMA complete	The transmit register is empty and DMA Tx has been completed	Read IIR

NOTE: The priority level with a lower number has the higher priority.

11.2.5 UARTn_LCR: UART n line control register

UARTn_LCR is an 8-bit register.

UART0_LCR=0x4000_800C, UART1_LCR=0x4000_810C, UART2_LCR=0x4000_820C
UART3_LCR=0x4000_830C

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	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN
-	0	0	0	0	0	11
-	RW	RW	RW	RW	RW	RW

6	BREAK	When this bit is set to 1, the TxD pin is driven low to alert the receiver.	
		0	General communication mode
		1	Transmit break mode

BREAK is a break control bit. If BREAK = 1, the serial output pin (TxD pin) is driven low (spacing state), whereas writing a 0 to the bit disables the break condition.

5	STICKP	Whether or not to use stick parity. The setting of this bit is valid only when the PEN bit is set to 1.	
		0	Does not use stick parity
		1	Uses stick parity

STICKP Determines whether or not to use stick parity. If PEN = 1, PARITY = 1, and STICKP = 1, the parity bit is always set low. If PEN = 1, STICKP = 1, and PARITY = 0, the transmit parity bit is always set high. If STICKP = 0, stick parity is disabled.

4	PARITY	Parity mode selection	
		0	Odd-parity mode
		1	Even-parity mode

PARITY Determines which parity mode is used between odd and even parity. If PEN = 1 and PARITY = 0, odd parity is used depending upon the number of data bits; if PEN = 1 and PARITY = 1, even parity is used.

3	PEN	Whether or not to enable parity	
		0	Disables parity.
		1	Enables parity.

PEN Determines whether or not to enable the use of parity. If PEN = 1, the transmitter creates a parity bit between the last data bit and the stop bit(s) and the receiver inspects this parity bit. (The parity bit is set to 0 or 1 to ensure that the total number of 1s in the data and parity bits is either even or odd depending on how the PARITY and STICKP bits are set.

2	STOPBIT	The number of stop bits demanded by the preceding data bits	
		0	The number of stop bits is 1.
		1	The number of stop bits is 1.5 or 2. If the data word is 5 bits long, 1.5 stop bits are added. If the data word is 6, 7, or 8 bits long, 2 stop bits are added.

STOPBIT Defines the number of stop bits attached to the end of each transmitted/received data word. STOPBIT = 0 includes 1 stop bit at the end of each transmitted data word. STOPBIT = 1 includes 1.5 or 2 stop bits at the end of each transmitted data word depending upon the number of transmit bits defined at DLEN; a 5-bit data word includes 1.5 stop bits and a 6-, 7-, or 8-bit data word includes 2 stop bits. The receiver checks only the first stop bit regardless of the number of stop bits defined at DLEN and STOPBIT.

0	DLEN	The length of data bits included by each data transfer	
		00	5-bit data
		01	6-bit data
		10	7-bit data
		11	8-bit data

DLEN Defines the length of each transmitted/received data word.

A parity bit is created based on the settings of LCR bits 3, 4, and 5 (PEN, PARITY, and STICKP). Table 48 illustrates various configurations to create a parity bit.

Table 48. Various Configurations to Create a Parity Bit

STICKP	PARITY	PEN	Parity
X	X	0	No parity
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Forced 1 stick parity
1	1	1	Forced 0 stick parity

11.2.6 UARTn_DCR: UART n data control register

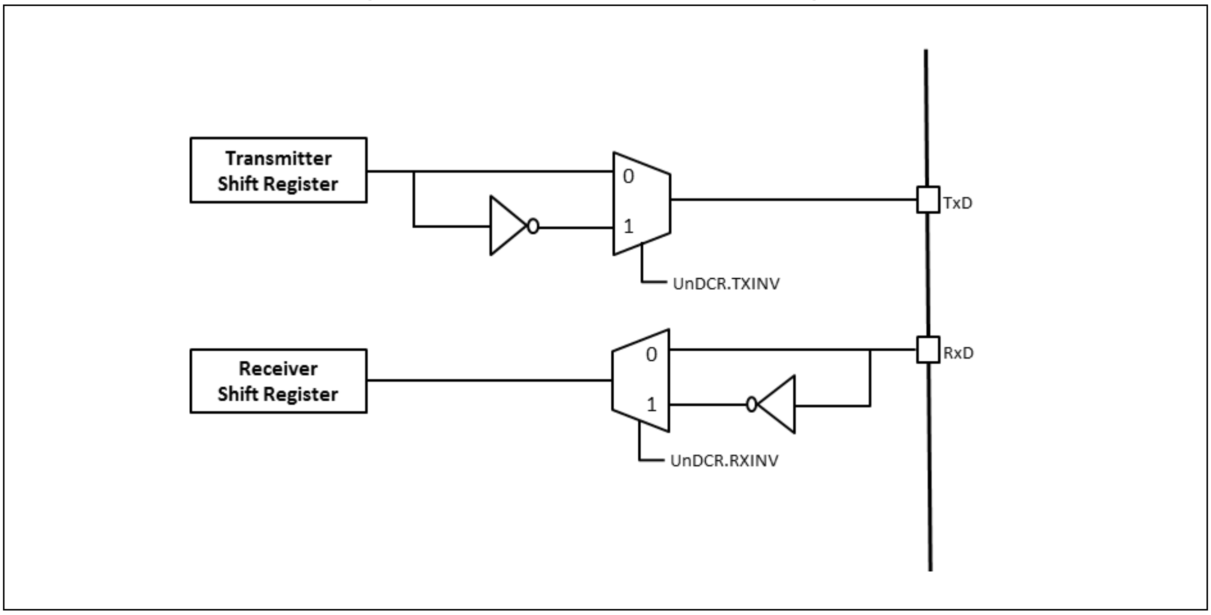
UARTn_DCR is a 32-bit register that controls Tx or Rx data inversion.

UART0_DCR=0x4000_8010, UART1_DCR=0x4000_8110, UART2_LCR=0x4000_8210
UART3_LCR=0x4000_8310

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	RXINV	TXINV	Reserved
-	0	0	-
-	RW	RW	-

3	RXINV	Rx data inversion selection	
		0	Inputs normal Rx data.
		1	Inputs inverted Rx data.
2	TXINV	Tx data inversion selection	
		0	Outputs normal Tx data.
		1	Outputs inverted Tx data.

Figure 65. Data Inversion Control Diagram



11.2.7 UARTn_LSR: UART n line status register

UARTn_LSR is a read-only register that shows the status of data transmission and reception.

UART0_LSR=0x4000_8014, UART1_LSR=0x4000_8114, UART2_LSR=0x4000_8214
UART3_LSR=0x4000_8314

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				TEMT	THRE	BI	FE	PE	OE	DR
-				1	1	0	0	0	0	0
-				RO	RO	RO	RO	RO	RO	RO

6	TEMT	Whether or not the transmit registers are empty	
		0	Data is being transmitted through the transmit registers.
		1	The transmit registers are empty.

TEMP Indicates whether or not the transmit registers are empty. When both UARTn_THR and TSR (Transmit Shift Register) are empty, the bit is set to 1. Once either THR or TSR holds any data, the bit is cleared to 0.

5	THRE	Whether or not the transmit data hold register is empty	
		0	The transmit data hold register is not empty.
		1	The transmit data hold register is empty.

THRE Indicates whether or not UARTn_THR is empty, which means the UART is ready to receive new data from the CPU for transmission. Empty interrupts can be generated if they are set enabled in UARTn_THR. The bit is set to 1 when new data can be written to UARTn_THR as the previous transmit data has been transferred from UARTn_THR to the transmit shift register (TSR). The bit is cleared to 0 when new data is written to UARTn_THR by the CPU (when THR is no longer empty).

4	BI	Whether or not a break condition has been detected	
		0	Normal.
		1	A break condition has been detected.

BI Shows the occurrence of a break interrupt. If the incoming receive data remains in “L” for longer than the time taken to receive the entire data word (i.e., the sum of the start, data, parity, and stop bits), the bit is set to 1. Once UARTn_LSR is read by the MCU, the bit is cleared to 0.

3	FE	Presence of a frame error	
		0	No frame error present.
		1	A frame error has been detected. The received data does not have appropriate stop bits.

FE Indicates whether or not a communication framing error has been detected. A framing error signifies that a received data word does not have an appropriate stop bit. The FE bit is set to 1, when the stop bit attached to the last data bit or the parity bit is detected to be 0. Once UARTn_LSR is read by the MCU, the bit is cleared to 0.

2	PE	Presence of a parity error	
		0	No parity error present
		1	A parity error has been detected. The received data does not satisfy parity conditions.

PE Indicates whether or not a parity error has been detected. If a received data word does not meet the odd/even parity condition defined in LCR, the bit is set to 1. Once UARTn_LSR is read by the MCU, the bit is cleared to 0.

1	OE	Presence of an overrun error	
		0	No overrun error present.
		1	An overrun error has been detected. Additional data has arrived when the RBR is already full.

OE Indicates whether or not an overrun error has been detected. When a data word is transferred to UARTn_RBR before the register reads the previous data word, an error occurs and the OE bit is set

to 1. Once UARTn_LSR is read by the MCU, the overrun error is not detected and the bit is cleared to 0.			
0	DR	Data reception status	
		0	There is no data in the receive data hold register.
		1	Data is received and stored in the receive data hold register.
DR Indicates that data reception has been completed. The bit is set to 1 when a data word has completely been received and transferred to UARTn_RBR. Once the data is read by UARTn_RBR, the bit is cleared to 0.			

This register reports the status of data transfers between the transmitter and receiver. Through this register, you can check the status of the UART lines and set interrupts as follows: Status interrupts for bits 1, 2, 3, and 4 can be called if they are set enabled at the UARTn_IER register’s RLSIE bit. Other bits enable their corresponding interrupts to be called if they are set enabled in the UARTIn_IER register.

11.2.8 UARTn_BDR: UART n baud-rate divisor latch register

UARTn_BDR is a 16-bit register.

UART0_BDR=0x4000_8020, UART1_BDR=0x4000_8120, UART2_BDR=0x4000_8220
UART3_BDR=0x4000_8320

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		BDR	
-		0x0000	
-		RW	

15 0	BDR	Baud-rate divisor latch

NOTE: When the BDR bits of the UARTn_BDR register is set to range from 0 to 2, the BFR bit of the UARTn_BFR register is not reflected to the baud rate settings.

To establish communication via a UART channel, you must set an appropriate baud rate. To this end, a programmable baud-rate generator is built in, providing a baud-rate divisor ranging from 1 to 65535. You must write an appropriate divisor value to the 16-bit BDR to obtain the desired baud rate.

Below is the formula for calculating the baud rate:

$$\mathrm{BDR} = \frac{MCCR7}{16 \times\ BaudRate}$$

If you have set the UART clock in the MCCR7, the PCLK clock speed must satisfy the condition of PCLK > MCCR7 x 2. (**NOTE:** The UART uses not PCLK but MCCR7 clock.)

Table 49 lists the divisor and error rate for each baud rate when the PCLK speed is 72MHz.

Table 49. Examples of Baud Rate Calculation

PCLK=72MHz		
Baud rate	**Divisor (BDR)**	**Error (%)**
1200	1875	0.00%
2400	937	0.05%
4800	468	0.16%
9600	234	0.16%
19200	117	0.16%
38400	58	1.02%
57600	39	0.16%
115200	19	2.79%

11.2.9 UARTn_BFR: UART n baud-rate fraction counter register

UARTn_BFR is an 8-bit register.

UART0_BFR=0x4000_8024, UART1_BFR=0x4000_8124, UART2_BFR=0x4000_8224
UART3_BFR=0x4000_8324

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		BFR
	-		0x00
	-		RW

7 0	BFR	Whether to enable or disable the fraction counter to compensate error from the integer baud-rate divisor	
		0	Disables the fraction counter.
		N	Enables the fraction counter. In fractional compensation mode, the value of the fraction counter is added to the integer baud-rate divisor.

Table 50. Examples of Baud Rate Calculation Using a BFR Value

PCLK=72MHz			
Baud rate	**Divisor (BDR)**	**FCNT (BFR)**	**Error (%)**
1200	1875	0	0.0%
2400	937	128	0.0%
4800	468	192	0.0%
9600	234	96	0.0%
19200	117	48	0.0%
38400	58	152	0.0%
57600	39	16	0.0%
115200	19	136	0.0%

The 8-bit fraction counter compensates the fractional part of the actual baud-rate divisor. This is needed because the baud-rate divisor latch register indicates the integer part of the baud-rate divisor only. By adding the fractional part indicated by UARTn_BFR to the integer part indicated by UARTn_BDR, the correct divisor is obtained to calculate the accurate baud rate.

For example, when the baud rate is 9600 bps:

$$\frac{PCLK\ /\ 2}{16\times\ BaudRate}=\frac{72000000\ /\ 2}{16\times\ 9600}=234.375\quad Divisor=234, Float=0.375$$

$$\mathrm{FCNT}=\mathrm{Float}*256\ =0.375*256=96$$

BDR = 234, BFR = 96

11.2.10 UARTn_IDTR: UART n inter-frame delay time register

UARTn_IDTR is an 8-bit register. A dummy delay is inserted between two consecutive transmit data frames.

UART0_IDTR=0x4000_8030, UART1_IDTR=0x4000_8130, UART2_IDTR=0x4000_8230
UART3_IDTR=0x4000_8330

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			SMS	DMS	Reserved			WAITVAL		
-			0	0	-			000		
-			RW	RW	-			RW		

7	SMS	Whether to enable or disable multi-sampling the start bit	
		0	Disables multi-sampling the start bit. At this setting, sampling is conducted only once at clock pulse 8 (among the 16 pulses in total).
		1	Enables multi-sampling the start bit. At this setting, sampling is conducted three times at clock pulses 7, 8, and 9 (among the 16 pulses in total). The value indicated by a majority of the three samples is taken.
SMS If this bit is set to 1, the start bit is sampled three times. The value indicated by a majority of the samples is taken. **NOTE:** After the MCU's power-on, if you disable the SCU_PER register's UART bit and then enable it again, the SMS bit is set to 1.			
6	DMS	Whether to enable or disable multi-sampling each data bit	
		0	Disables multi-sampling each data bit. At this setting, sampling is conducted only once at clock pulse 8 (among the 16 pulses in total).
		1	Enables multi-sampling each data bit. At this setting, sampling is conducted three times at clock pulses 7, 8, and 9 (among the 16 pulses in total). The value indicated by a majority of the three samples is taken.
DMS If this bit is set to 1, each data bit is sampled three times. The value indicated by a majority of the samples is taken.			
0	WAITVAL	Defines the wait time for the next data frame. [unit: 1-bit period]	
		$\text{Wait Time} = \frac{WAITVAL}{BAUDRATE}$	
WAITVAL Based on the value (0 through 7) indicated by the WAITVAL bits, a wait time is set between successive data transfers. The formula above is used to compute the wait time.			

11.3 Functional description

The UART module is compatible with 16450 UART. It also provides DMA channels and a fractional compensation logic to obtain baud rates. Because it does not include a FIFO block, data transfers are performed either interactively or by means of DMA support. The DMA operates as follows:

A UART can be linked with two DMA channels, of which one is responsible for Tx transfers and the other is for Rx transfers. Each channel has a 32-bit memory address register and a 16-bit transfer counter register. You must set these two registers before enabling the DMA. The memory address for Rx transfers is the target address, whereas the address for Tx transfers is the source address.

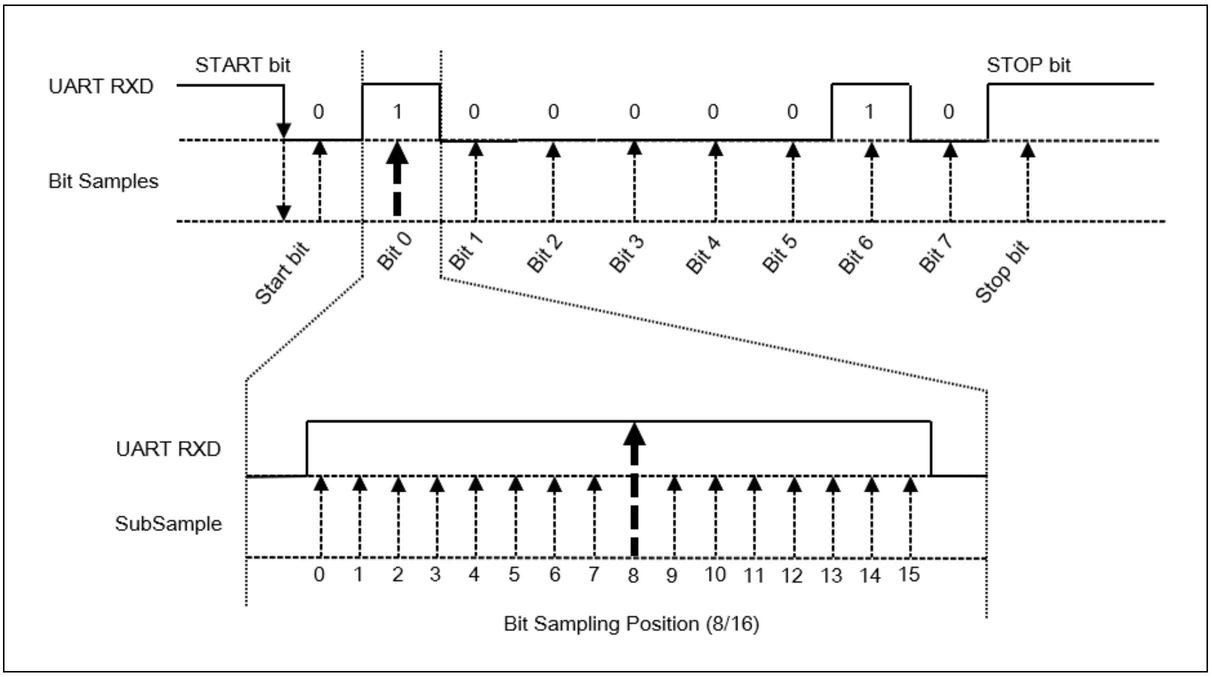
The transfer counter register records the number of transfers. The counter decreases by 1 every time a single-ended transfer is completed. When the counter reaches zero, a DMA complete flag is sent to the UART control block. Then, the corresponding interrupt is flagged if the interrupt has been set enabled.

11.3.1 Receive data sampling timing

The timing of UART operation is as follows:

Once a falling edge is detected in the receive line, the UART determines that a start bit is being received. The UART oversamples the received data 16 times per bit beginning from the start bit. Among the 16 samples, the value at the seventh clock pulse is determined to represent the value of the bit.

Figure 66. Sampling Timing of UART Receiver



To enhance protection against external glitch noises, it is recommended to enable debouncing in the PCU block.

11.3.2 Transmit data format

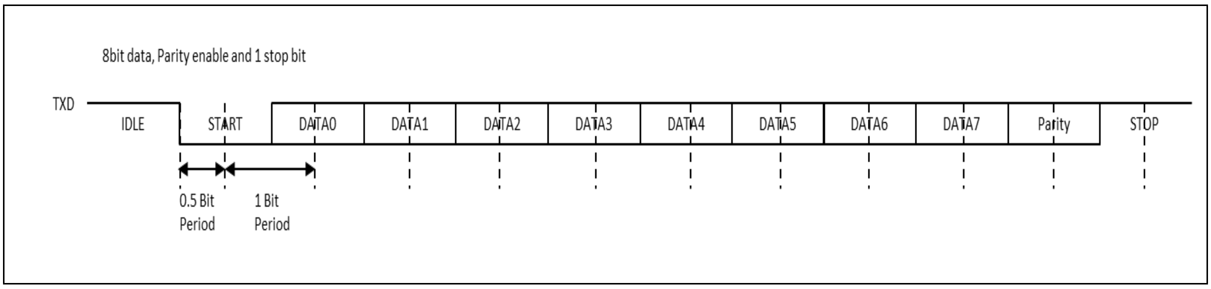
The UART transmitter is in charge of transmitting data. For each data word, the start, data, optional, and stop bits are shifted serially from the least significant bit.

The number of data bits is defined in the UARTn_LCR's DLAN[1:0] bit.

The parity bit type is configured with UARTn_LCR's PARITY and PEN bits. If even parity is selected, the parity bit is determined by the bit sum of all the data bits. For odd parity, the parity bit takes the opposite value to that of even parity. The number of stop bits is defined at the UARTn_LCR's STOPBIT bit.

Below is a transmit data format example (Out of the 16 samples taken from 1 bit, the seventh sample represents the value of the bit):

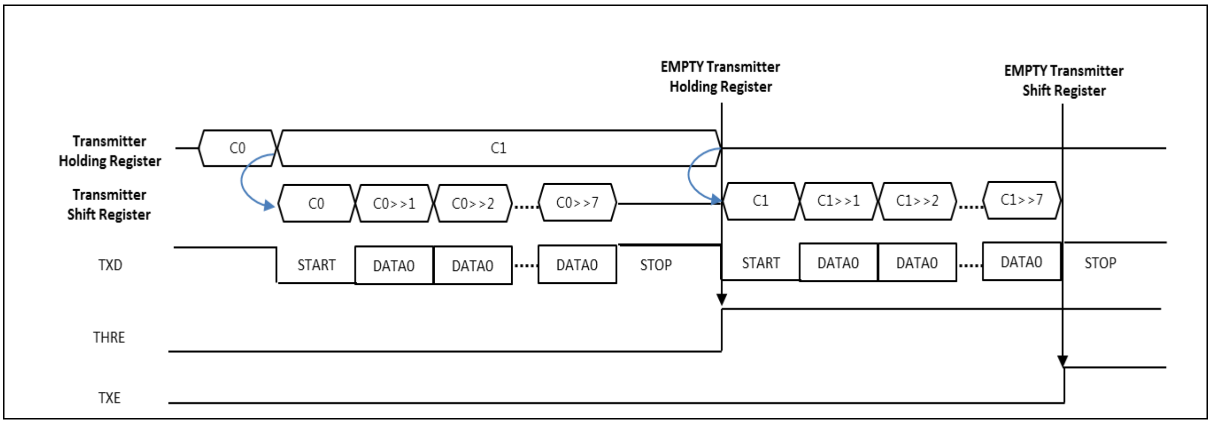
Figure 67. Transmit Data Format Example



11.3.3 Transmit interrupt

Transmitting uses some types of interrupt flags. When the transmit data hold resistor (THR) is empty, the THRE interrupt flag is set to 1; when the transmit shift resister (TSR) is empty, the TXE interrupt flag is set to 1. The user can select the interrupt timing that most suits their application.

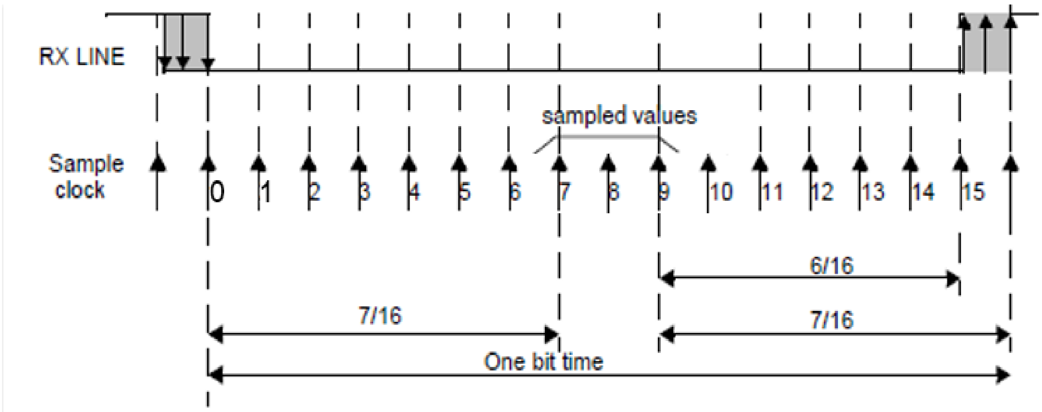
Figure 68. Transmit Interrupt Timing Diagram


EMPTY Transmitter Holding Register
EMPTY Transmitter Shift Register
Transmitter Holding Register
C0
C1
Transmitter Shift Register
C0
C0>>1
C0>>2
C0>>7
C1
C1>>1
C1>>2
C1>>7
TXD
START
DATA0
DATA0
DATA0
STOP
START
DATA0
DATA0
DATA0
STOP
THRE
TXE

11.3.4 Multi-sampling

If UARTn_IDTR’s SMS or DMS bit is set to 1, one of the values sampled at the 7th, 8th, and 9th clock pulses is determined to represent the value of the receive data bit.

Figure 69. Multi-sampling Timing of Start and Data Bits of Receive Data

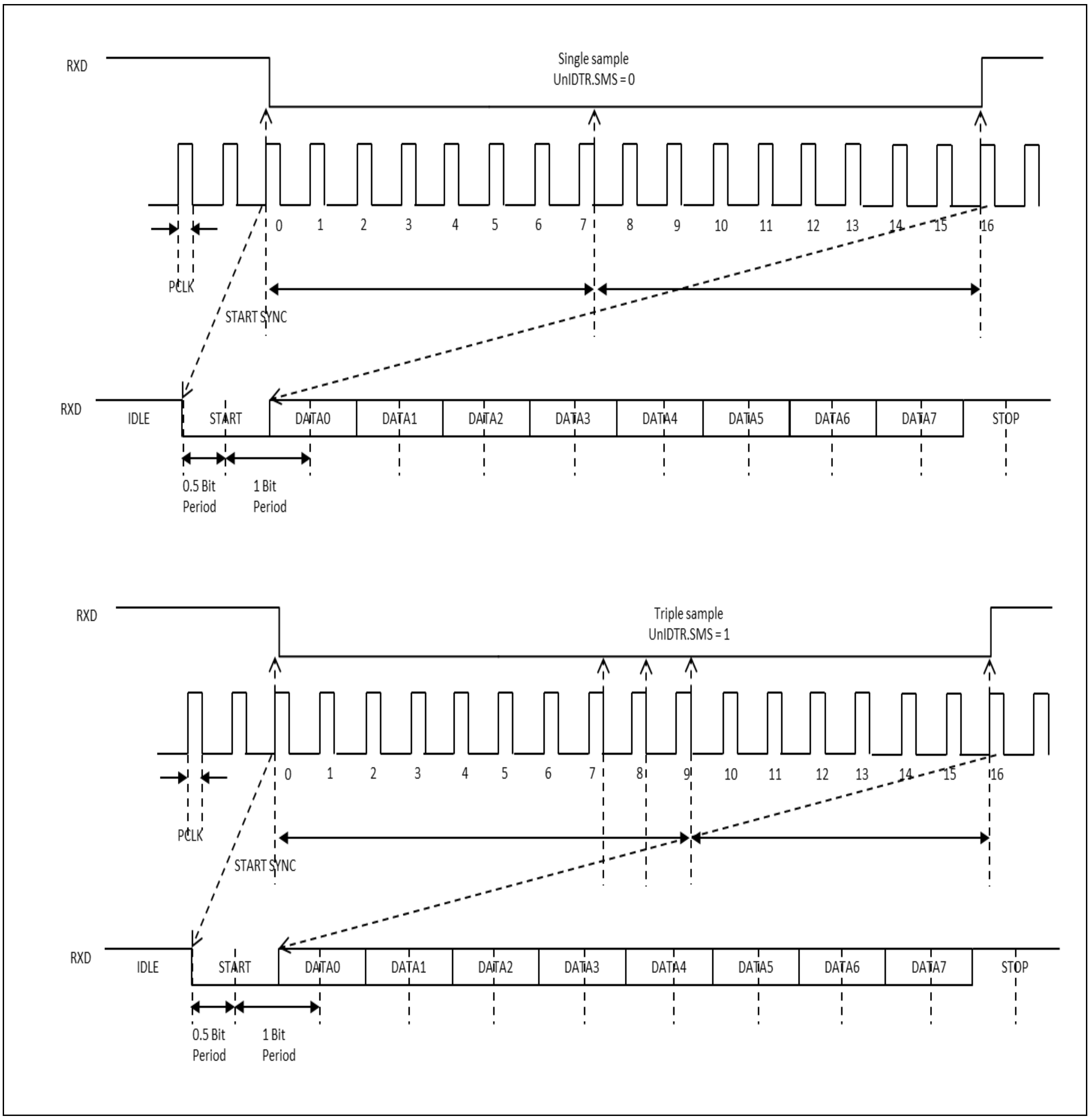


11.3.5 Start bit detection

On the falling edge of the start bit is a digital filter that rejects noise signals. A pulse shorter than (3 * (1/baud rate)) / 16 sec is rejected by this filter. Even when a pulse is longer than (3 * (1/baud rate)) / 16 sec, it can be rejected under the multi-sampling strategy during start-bit sampling.

Below are timing diagrams for single- and multi-sampling modes.

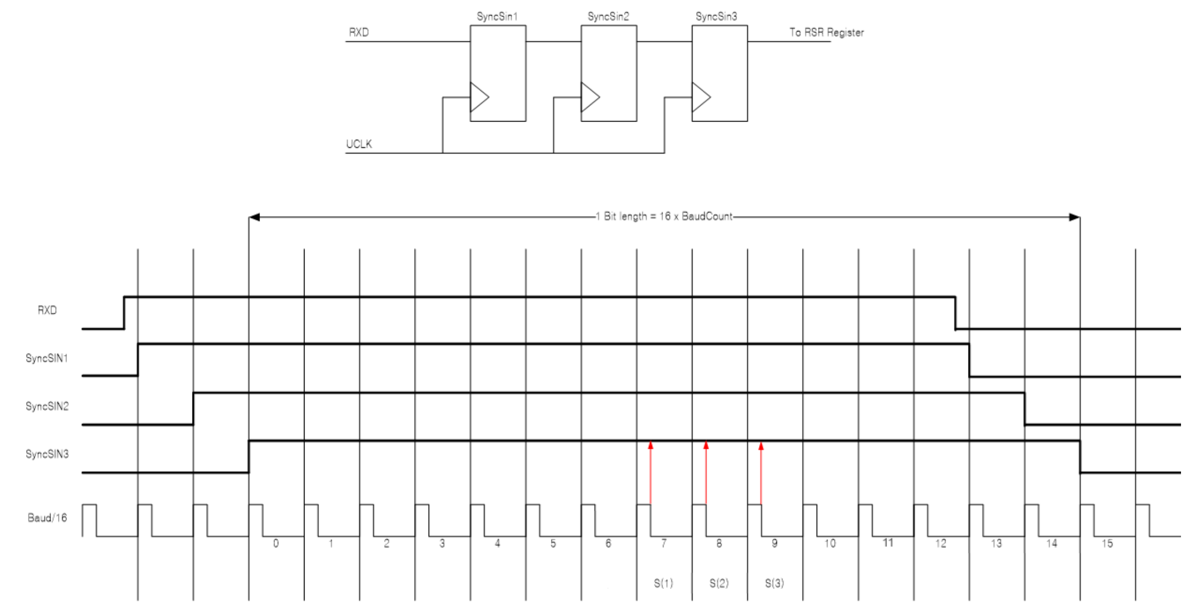
Figure 70. Start Bit Detection and Single-/Multi-sampling Timing



11.3.6 Data sampling strategy

An internal synchronous circuit is used in the receiver block to prevent abnormal noises in the RXD signal line. The start and data bits can be either single-sampled or multi-sampled. The UARTn_IDTR register’s IDTR.SMS bit defines the sampling strategy for the start bit, and the IDTR.DMS bit defines the sampling strategy for the data bits.

Figure 71. Data Sampling Timing



The figure above shows a diagram of the anti-noise synchronous circuit and sampling timing.

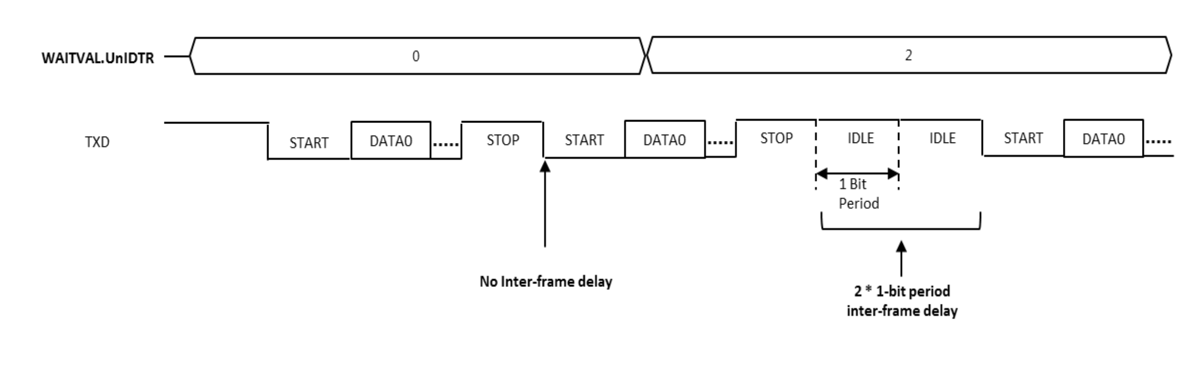
If the DMS bit is set enabled, the value of each bit is sampled at multiple clock pulses. For the length of each data bit, the 9th samples at pulses S(1), S(2), and S(3) are taken out of 16 samples to determine the bit’s value by taking the value indicated by a majority of the samples.

If the DMS bit is set disabled, on the other hand, the single-sampling process takes the value at S(2) only. The DMS bit does not affect the start bit, but the SMS bit works in a similar way for the start bit.

11.3.7 Inter-frame delay for data transmission

The inter-frame delay functionality creates an idle state between two characters on the TxD line. The length of the idle state can be defined by the IDTR register's WAITVAL bits. If WAITVAL is set to 0, a delay does not occur; otherwise, the transmitter waits for the number of bit periods defined at the WAITVAL bit field after transmitting the "high" part of TxD.

Figure 72. Transmit Data Format Example



11.3.8 DMA transfers

The UART module supports DMA transfer by using the DMA device built in the MCU. The start of the memory address for data transfer and the length of the data to be transferred are determined by programming registers in the DMA block. The completion of data transfer is related to a notification by the transfer complete flag. Once the entire transmit data is written to the transmit data hold register (THR), the UARTn_IIR.DMA TX TXE bit (DMA Tx complete) is flagged along with the ID of the complete interrupt written to the register.

Once the entire receive data is written to the target DMA memory, the UARTn_IIR register's DMA Rx complete flag is set along with the ID of the complete interrupt written to the IIR register.

Therefore, the UART RxD signal already became idle when the DMA Rx complete interrupt occurred.

11.3.9 Setting examples

<Example 1> UART0 initial setting (baud rate = 19,200 bps, PCLK = 8MHz)

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                    : Unlocks the SCU registers.
SCU_PER2<UART0[8]> = “1”                            : Enables the UART0 peripheral.
SCU_PCER2<UART0[8]> = “1”                           : Enables the UART0 peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”                    : Enables PORTEN (enter 0x15 and then 0x51).
PCU_PORTEN<PORTEN[7:0] = “0x51”

PC_MR2<P14MUX[26:24]> = “01”                        : Sets Port C pin P14 as RXD0.
PC_MR2<P15MUX[30:28]> = “01”                        : Sets Port C pin P15 as TXD0.
PC_CR<P14[29:28]> = “10”                            : Sets Port C pin P14 as an input pin.
PC_CR<P15[31:30]> = “00”                            : Sets Port C pin P15 as a push-pull output pin.

UARTn_DCR< RXINV[3]> = “0”                          : Enables normal Rx data input.
UARTn_DCR< RXINV[3]> = “0”                          : Enables normal Tx data output.
UARTn_LCR<DLEN[0:1]> = “11”                         : Sets the data length to 8 bits.
UARTn_LCR<STOPBIT[2]> = “0”                         : Sets the number of stop bits.
UARTn_LCR<PEN[3]> = “0”                             : Disables parity.
UARTn_LCR<PARITY[4]> = “0”                          : Selects odd-parity mode.
UARTn_BDR<DLL[7:0]> = “00001101”                    : Sets the baud rate to 19200 bps.
                                                     BDR = (8MHz/2)/(16x13) = approx.
                                                       19230.769 bps
UARTn_BFR<BFR[7:0]>= “00000101”                     : Sets the baud-rate fractional value register.
                                                     (8MHz/2)/(16x19200) = 13.020 Divisor = 13,
                                                       Float = 0.02
                                                     FCNT = 0.02x256 = 5.12
UARTn_RBR<RO[7:0]> READ                             : Reads the receive data byte.
UARTn_THR<THR[7:0]> READ                            : Reads the transmit data byte.
```

12 Serial Peripheral Interface (SPI)

A33M11x series has two Serial Peripheral Interface (SPI) modules built in. The SPI modules are synchronized by clocks. The specifications of the transmit and receive clocks are adjustable. The SPI supports communications between one master and multiple slaves. Slaves can be selected using slave select (SS).

The SPI performs four-wire synchronous transfers via four signal terminals (SS, SCK, MOSI, and MISO). Its separate transmit and receive buffers enables full-duplex communication, which is capable of reading and writing data simultaneously.

The SPI of A33M11x series features the followings:

- Selectable between the master and slave operations
- Full-duplex and four-wire synchronous transfers supported
  - SS: Slave Select
  - SCLK: Serial Clock
  - MOSI: Master Output Slave Input
  - MISO: Master Input Slave Output
- SPI clock speed and polarity adjustable
- Separate transmit and receive data registers with different data transfer sizes
  - Available transmit/receive data sizes: 8, 9, 16, and 17 bits
- Configurable interrupts triggered by the transmit status and SS signal
- Loop-back mode for internal checkups
- User-programmable start, burst, and stop delay times
- DMA transfers

Table 51 introduces pins assigned for SPI.

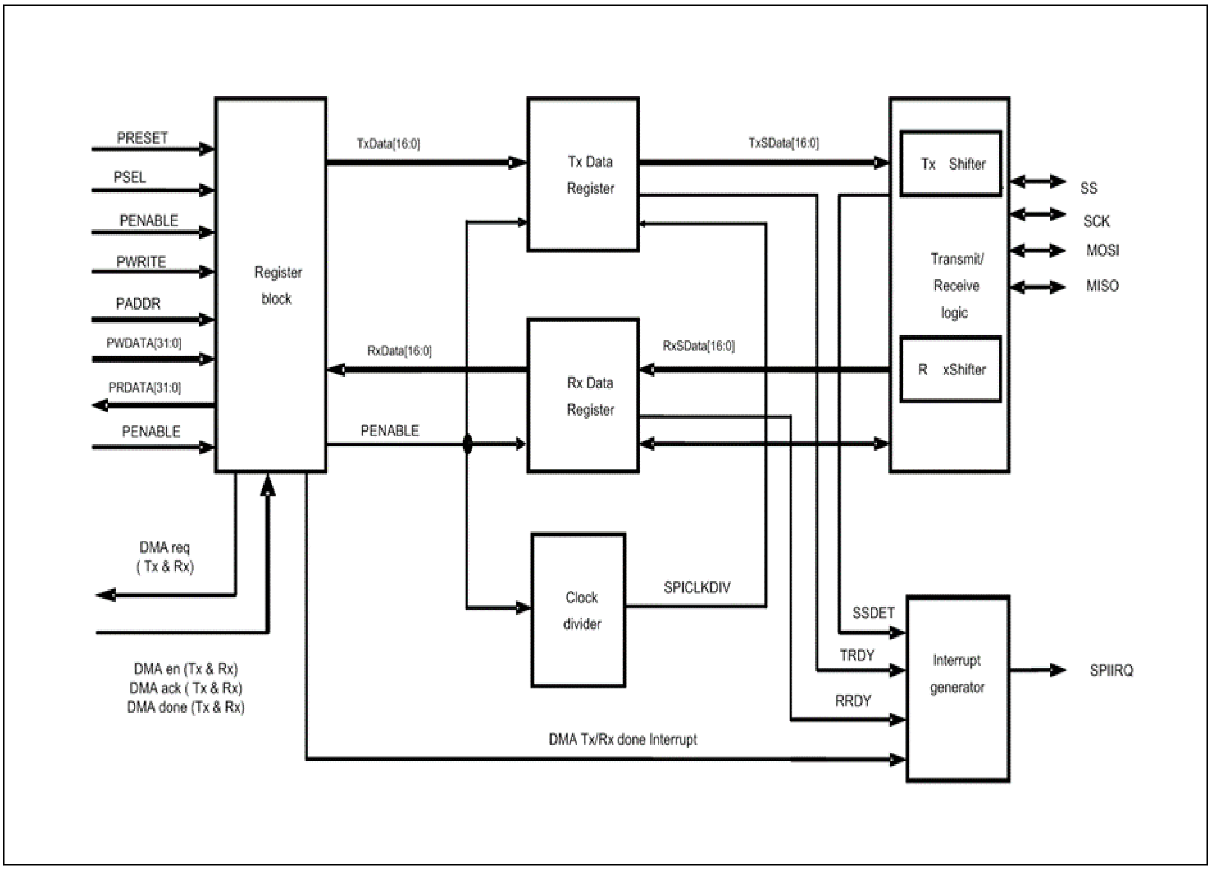
Table 51. Pin Assignment of SPI: External Pins

Pin name	Type	Description	Supported Packages	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
SS0	I/O	SPI0 serial port I/O signal for slave selection	O	O
SCK0	I/O	SPI0 clock I/O (master: output / slave: input)	O	O
MOSI0	I/O	SPI0 transmit and receive data (master: output / slave: input)	O	O
MISO0	I/O	SPI0 transmit and receive data (master: input / slave: output)	O	O
SS1	I/O	SPI1 serial port I/O signal for slave selection	O	O
SCK1	I/O	SPI1 clock I/O (master: output / slave: input)	O	O
MOSI1	I/O	SPI1 transmit and receive data (master: output / slave: input)	O	O
MISO1	I/O	SPI1 transmit and receive data (master: input / slave: output)	O	O

12.1 SPI block diagram

In this section, SPI is described in a block diagram in Figure 73.

Figure 73. SPI Block Diagram



12.2 Registers

Base address of SPI is introduced in the followings:

Table 52. Base Address of SPI

Name	Base address
SPI0	0x4000_9000
SPI1	0x4000_9100

Table 53. SPI Register Map

Name	Offset	Type	Description	Reset value	Reference
SPIn_TDR	0x0000	WO	SPI n transmit data register	-	12.2.1
SPIn_RDR	0x0000	RO	SPI n receive data register	0x0000_0000	12.2.2
SPIn_CR	0x0004	RW	SPI n control register	0x0000_1020	12.2.3
SPIn_SR	0x0008	RC	SPI n status register	0x0000_0006	12.2.4
SPIn_BR	0x000C	RW	SPI n baud-rate register	0x0000_FFFF	12.2.5
SPIn_EN	0x0010	RW	SPI n enable register	0x0000_0000	12.2.6
SPIn_LR	0x0014	RW	SPI n delay length register	0x0001_0101	12.2.7

NOTE: n = 0 and 1

12.2.1 SPIn_TDR: SPI n transmit data register

SPIn_TDR is a 17-bit sized read/write register. It contains serial transmit data.

SPI0_TDR=0x4000_9000, SPI1_TDR=0x4000_9100

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td colspan="17">TDR</td></tr>
<tr><td colspan="15">-</td><td colspan="17">0x00000</td></tr>
<tr><td colspan="15">-</td><td colspan="17">WO</td></tr>
</table>

16 0	TDR	Transmit Data Register

12.2.2 SPIn_RDR: SPI n receive data register

SPIn_RDR is a 17-bit sized read/write register. It contains serial receive data.

SPI0_RDR=0x4000_9000, SPI1_RDR=0x4000_9100

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td colspan="17">RDR</td></tr>
<tr><td colspan="15">-</td><td colspan="17">0x00000</td></tr>
<tr><td colspan="15">-</td><td colspan="17">RO</td></tr>
</table>

16 0	RDR	Receive Data Register

12.2.3 SPIn_CR: SPI n control register

SPIn_CR is a 20-bit sized read/write register and can be set to configure SPI operation mode.

SPI0_CR=0x4000_9004, SPI1_CR=0x4000_9104

3 3 2 2 2 2 2 2 1 0 9 8 7 6 5 4	2 2 2 3 2 1	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	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-low clocks selected.
		1	Active-high 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

12.2.4 SPIn_SR: SPI n status register

SPIn_SR is a 10-bit sized read/write register. It contains the status of SPI interface.

SPI0_SR=0x4000_9008, SPI1_SR=0x4000_9108

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	SBUSY	SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
-																						0	0	0	0	0	0	0	1	1	0
-																						RC	RC	RO	RC	RC	RC	RC	RO	RO	RO

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.
7	SBUSY		Transmit or receive flag
		0	Idle state
		1	Transmit or receive in progress
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 SPIn_RDR.
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 SPIn_TDR.
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 SPIn_TDR.
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 SPIn_RDR.

12.2.5 SPIn_BR: SPI n baud rate register

SPIn_BR is a 16-bit sized read/write register. Baud rate can be set by writing the register.

SPI0_BR=0x4000_900C, SPI1_BR=0x4000_910C

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
-	0xFFFF
-	RW

15 0	BR	Baud rate setting bits Baud Rate = PCLK / (BR + 1) (BR must be bigger than "0", BR >= 2)

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

12.2.6 SPIn_EN: SPI n enable register

SPIn_EN is a bit sized read/write register. It contains SPI enable bit.

SPI0_EN=0x4000_9010, SPI1_EN=0x4000_9110

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	
		0	SPI is disabled. SPInSR is initialized by writing "0" to this bit but other registers aren't initialized.
		1	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 SPIn_TDR before this bit is active.

12.2.7 SPIn_LR: SPI n delay length register

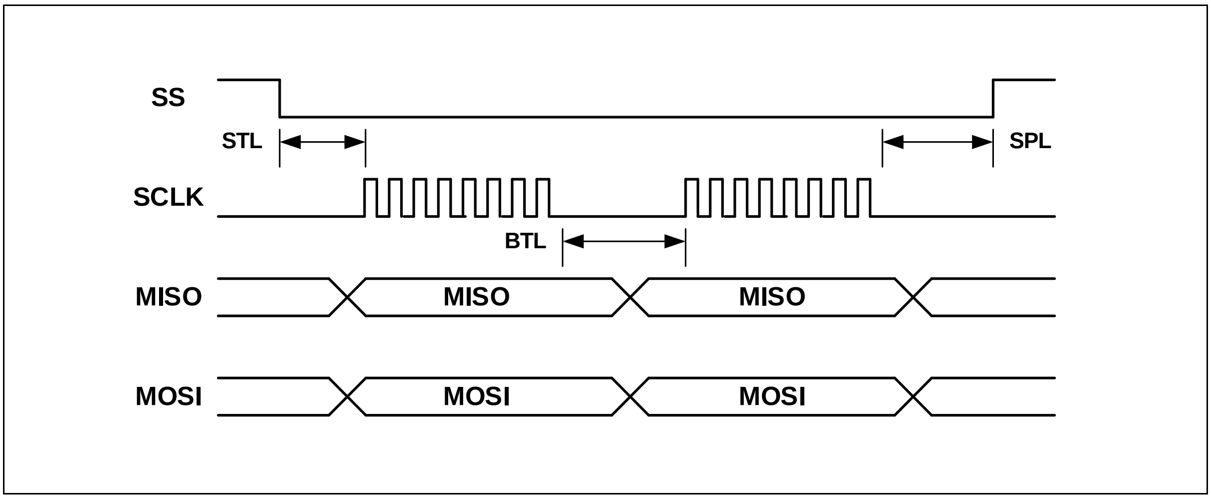
SPIn_LR is a 24-bit sized read/write register. It contains start, burst, and stop length value.

SPI0_LR=0x4000_9014, SPI1_LR=0x4000_9114

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 74. SPI Waveforms: STL, BTL, and SPL



NOTE Due to the nature of code Flash, the access time is slower than RAM, causing Flash waits. Since the reception of data during SPI high speed communication performed in code Flash may be faster than the data access time, so we recommend using a delay between the sending SPI data. By running code in SRAM and using DMA, you can use it without delay in high speed communication.

$$BTL(burst\ delay) \geq \frac{(Access\ wait + 1) * SCLK * (xbit + 2)}{HCLK} + \frac{4}{(BR + 1)}$$

12.3 Functional description

The transmitter and receiver of the serial peripheral interface (SPI) share the same clock but are independent of each other. This enables full-duplex transfers. The transmitter and receiver are equipped with double buffers, thereby supporting back-to-back data transfers either by reading previous receive data from the RDR register while subsequent data is being received, or by writing subsequent data to the TDR register while previous data is being transmitted.

To enable SPI transmission reception, the MOSI and MISO lines of the master and slave must be connected directly, enabling back-to-back transfers between them. The most significant bits are given priority during these data transfers. Communication is always commenced by the master. Once the master transfers data to the slave through pin MOSI, the slave responds through pin MISO.

12.3.1 Slave selection pin

The nSS line is used for slave select input that enables the slave to communicate with the master. The nSS pin can be driven as a standard IO port of the master device and is configured by setting the SSMOD bit in the SPI_CR register. If the SSMOD bit is set to 1, the pin is driven internally according to the setting of the SSOUT bit in the SPI_CR register. If the SSMOD bit is set to 0, the pin performs one of the two functions depending on the setting of the SSMO bit in the SPI_CR register.

When SSMO = 1 and SSMOD = 0, the nSS pin is used only in master mode. In this case, the nSS signal is driven low when the master starts communication. This low state is maintained until the SPI becomes disabled. When SSMO = 0 and SSMOD = 0, the multi-master function is enabled for all devices set in master mode. In slave mode, the nSS pin functions as an input pin. When the nSS signal is low, the device is selected as a slave; when it is driven high, the slave selection is released.

For the device in master mode, therefore, you can either set the nSS pin as an output pin to deactivate the pin (SSMOD = 0 , SSMO = 0); or set the pin as an input pin, feed it with a high-level signal (SSMOD = 1, SSOUT = 1), and, after a period of waiting time, activate the pin (SSMOD = 0, SSMO = 1); or feed the pin with a low-level signal (SSMOD = 1, SSOUT = 0) to generate a falling edge.

12.3.2 Clock phase and clock polarity

Each SPI has four operating modes for synchronizing data transferred through the MOSI and MISO lines with the SCK clock. The four modes determine the direction of data transfers and are set with the CPHA and CPOL bits in the SPI_CR register.

The CPHA (clock phase control) bit is used to select a transfer format among two different types. When the bit is set to 0, data is read at the first clock edge, which is an odd-numbered edge; when it is set to 1, data is read at the second clock edge, which is an even-numbered edge.

The CPOL (clock polarity control) bit is used to set the default level of the clock signal as active-high or active-low. This does not significantly affect the transfer format. Switching the bit causes the clock signal to be inverted (e.g., active-high is inverted to active-low; idle-low is inverted to idle-high).

To ensure appropriate communication between the master and slave devices, the two devices must be set in the same operating mode. Therefore, it might be needed to reconfigure the master device to fit the requirements of peripheral slaves.

Figure 75. SPI Transfer Timing 1/4: CPHA = 0, CPOL = 0, MSBF = 0

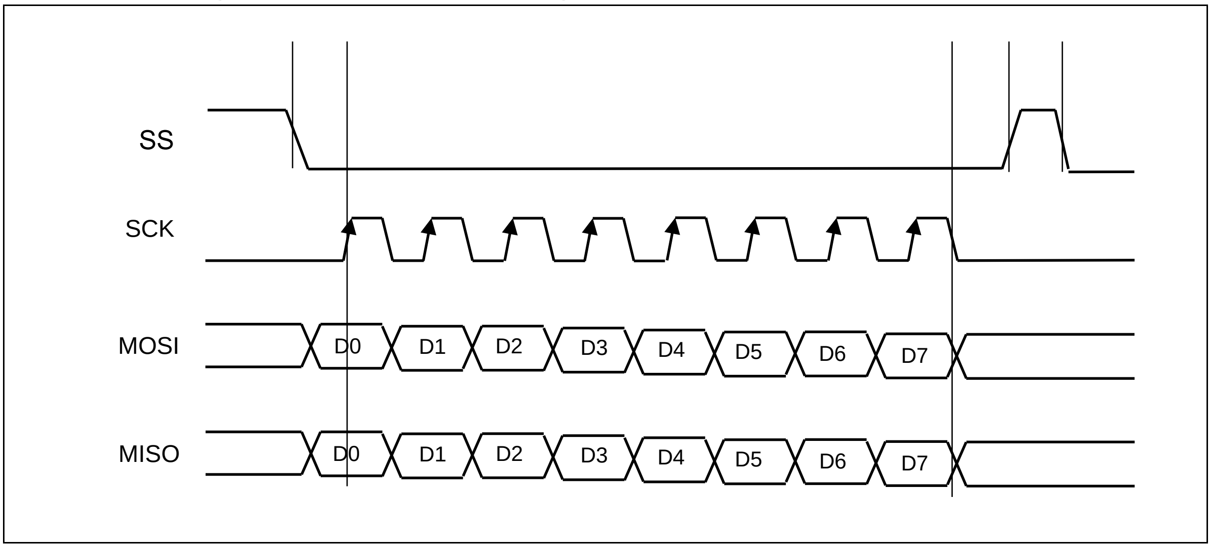


Figure 76. SPI Transfer Timing 2/4: CPHA = 0, CPOL = 1, MSBF = 1

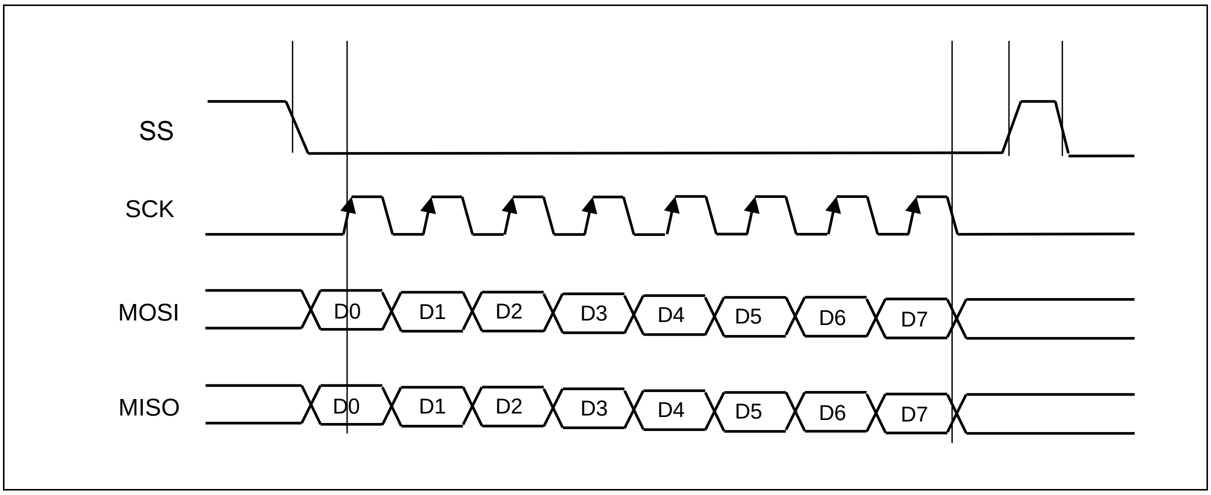


Figure 75 and Figure 76 illustrate SPI transfer timings when CPHA is set to 0. When CPHA = 0, the master and slave devices can read data at an odd-numbered (first) clock edge and change data at an even-numbered (second) clock edge.

CPOL is used to set the default level of the SCK clock signal. If CPOL = 0, the default level is set low; if CPOL = 1, the default level is set high.

The MSBF bit is used to select among the MSB- or LSB-first transfer modes for output from the MISO line. If MSBF = 1, data is shifted out bit by bit from the most significant bit down to the least significant bit; if MSBF = 0, data is shifted out bit by bit from the least significant bit down to the most significant bit.

Once all data has been transferred and the nSS pin is set to 1, the SCLK clock signal is no longer generated and the MISO and MOSI lines become high-Z.

Figure 77. SPI Transfer Timing 3/4: CPHA = 1, CPOL = 0, MSBF = 0

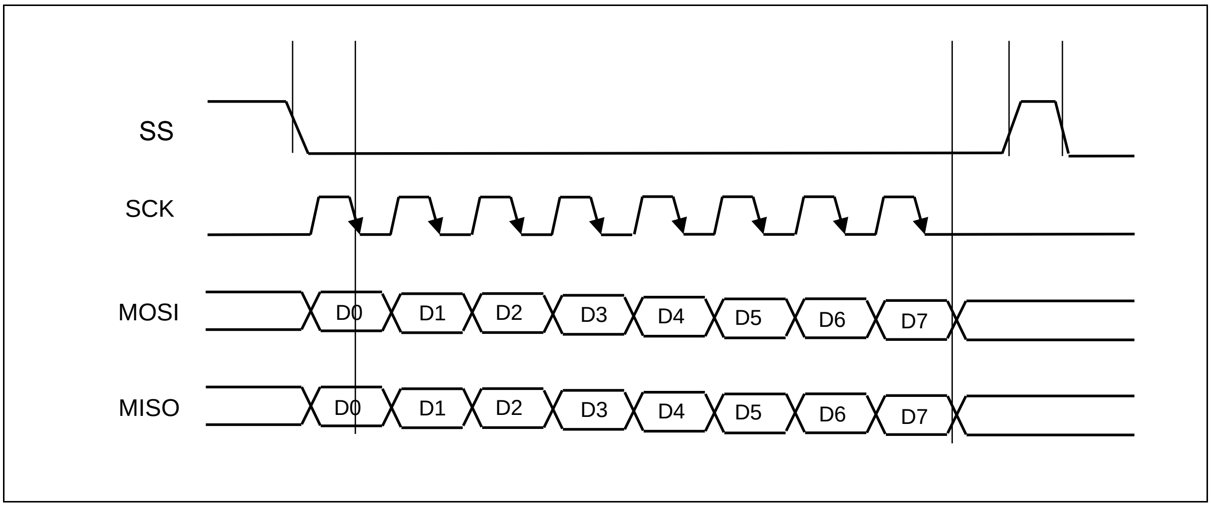


Figure 78. SPI Transfer Timing 4/4: CPHA = 1, CPOL = 1, MSBF = 1

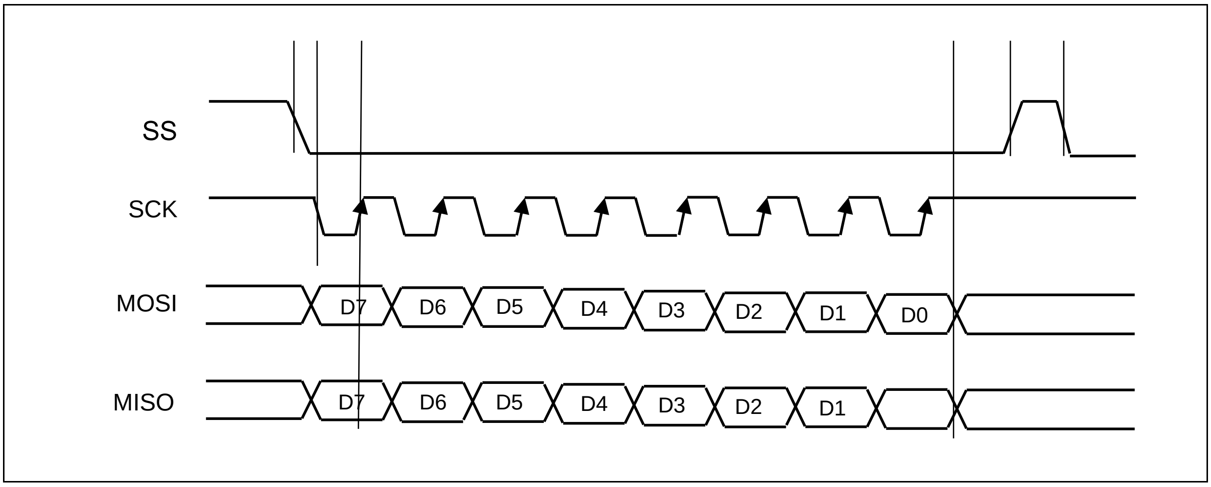


Figure 77 and Figure 78 illustrate SPI transfer timings when CPHA is set to 1. When CPHA = 1, the master and slave devices can read data at an even-numbered (second) clock edge and change data at an odd-numbered (first) clock edge.

CPOL is used to set the default level of the SCK clock signal. If CPOL = 0, the default level is set low; if CPOL = 1, the default level is set high.

The MSBF bit is used to select among the MSB- or LSB-first transfer modes for output from the MISO line. If MSBF = 1, data is shifted out bit by bit from the most significant bit down to the least significant bit; if MSBF = 0, data is shifted out bit by bit from the least significant bit down to the most significant bit.

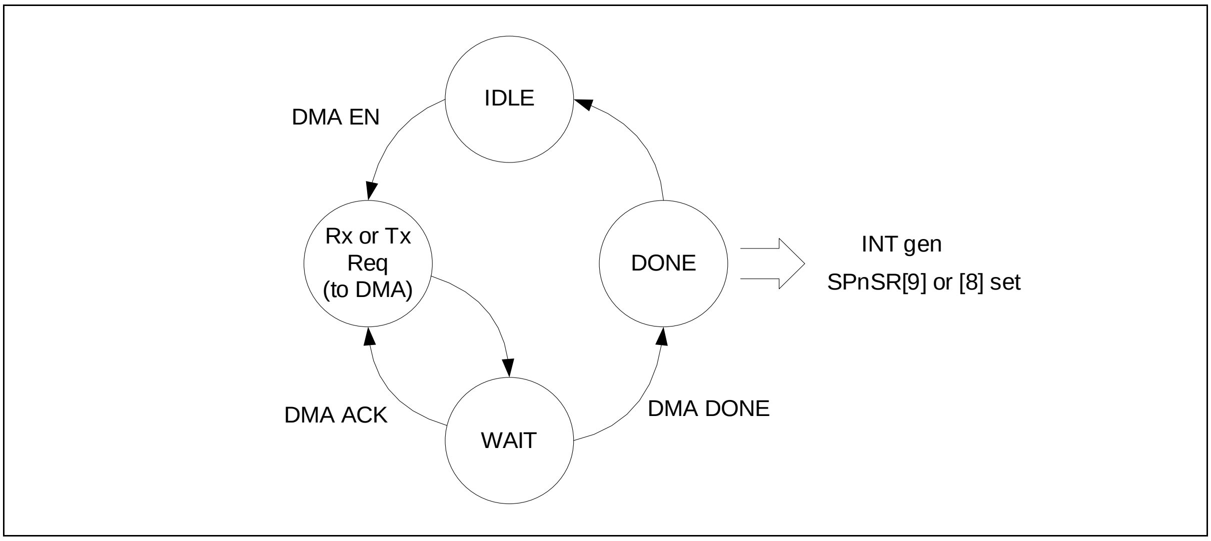
Once all data has been transferred and the nSS pin is set to 1, the SCLK clock signal is no longer generated and the MISO and MOSI lines become high-Z.

12.3.3 DMA handshake

The SPI supports the DMA handshaking operation. In order to operate a DMA handshake, DMA registers should first be set (refer to Chapter 8, Direct Memory Access Controller). SPIs use two DMA channels. As the transmitter and receiver are independent of each other, the SPI can operate the two channels at the same time. Once the DMA channel for the receiver is enabled and the receive buffer is filled, the SPI sends an Rx request to the DMA to empty the buffer and waits for an acknowledge signal from DMA. If the receive buffer is filled again after the acknowledge signal, the SPI sends an Rx request. If DMA Rx DONE becomes high, RXDMAF (SPInSR[8]) becomes 1 and an interrupt is serviced when RXDIE (SPInCR[17]) is set.

Similarly, if the transmit buffer is empty after the DMA channel for the transmitter is enabled, the SPI sends a Tx request to the DMA to fill the buffer and waits for an acknowledge signal from DMA. If the transmit buffer is empty again after the acknowledge signal, the SPI sends a Tx request. If DMA Tx DONE becomes high, TXDMAF (SPInSR[9]) becomes 1 and an interrupt is serviced when TXDIE (SPInCR[18]) is set. The slave transmitter sends dummy data at the first transfer (for 8–17 SCLK pulses) in DMA handshake mode.

Figure 79. DMA Handshake Workflow



12.3.4 Setting examples

<Example 1> Initial configuration of master SPI0

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                  : Unlocks the SCU registers.
SCU_PER2<SPI0[0]> = “1”                           : Enables the SPI0 peripheral.
SCU_PCER2<SPI0[0]> = “1”                          : Enables the SPI0 peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”                  : Enables PORTEN (enter 0x15 and then 0x51).
PCU_PORTEN<PORTEN[7:0] = “0x51”

PA_MR<P12MUX[18:16]> = “001”                      : Sets PA12 as SS0.
PB_MR<P13MUX[22:20]> = “001”                      : Sets PA13 as SCK0.
PB_MR<P14MUX[26:24]> = “001”                      : Sets PA14 as MOSI0.
PB_MR<P15MUX[30:28]> = “001”                      : Sets PA15 as MISO0.
PB_CR<P12[25:24]> = “00”                          : Sets PA12 (SS0) as an output pin.
PB_CR<P13[27:26]> = “00”                          : Sets PA13 (SCK0) as an output pin.
PB_CR<P14[29:28]> = “00”                          : Sets PA14 (MOSI0) as an output pin.
PB_CR<P15[31:30]> = “10”                          : Sets PA15 (MISO0) as an input pin.
PB_PRCR<PUE12[25:24]> = “00”                      : Disables pull-up/pull-down at PA12 (SS0).
PB_PRCR<PUE13[27:26]> = “00”                      : Disables pull-up/pull-down at PA13 (SCK0).
PB_PRCR<PUE14[29:28]> = “00”                      : Disables pull-up/pull-down at PA14 (MOSI0).
PB_PRCR<PUE15[31:30]> = “10”                      : Enables pull-up/pull-down at PA15 (MISO0).

SPI0_CR<BITSZ[1:0]> = “00”                        : Sets the transmit-receive bit size to 8 bits.
SPI0_CR<MSBF[4]> = “1”                            : Selects MSB-first transmit.
SPI0_CR<MS[5]> = “1”                              : Selects SPI0 as the master.
SPI0_CR<SSPOL[8]> = “1”                           : Sets the SS0 output signal high.
SPI0_CR<CPHA[3]> = “1”                            : Outputs data at each shift of an inverted phase signal.
SPI0_CR<CPOL[2]> = “1”                            : Starts the clock for data sampling in the high state.

SPI0_BR<BR[7:0]> = “0100 1111”                    : Sets the transmit-receive rate to 0.1 Mbps.
                                                   PCLK: 8MHz, PCLK/(79+1) = 0.1MHz
SPI0_EN<ENABLE[0]> = “1”                          : Sets the transmit buffer to be reset after the first data
SPI0_EN<ENABLE[0]> = “0”                          transmit.
```

<Example 2> Initial configuration of slave SPI1

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                    : Unlocks the SCU registers.
SCU_PER2<SPI0[0]> = “1”                             : Enables the SPI0 peripheral.
SCU_PCER2<SPI0[0]> = “1”                            : Enables the SPI0 peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”                    : Enables PORTEN (enter 0x15 and then 0x51).
PCU_PORTEN<PORTEN[7:0] = “0x51”

PB_MR1<P10MUX[10:8]> = “010”                        : Sets PB10 as SS1.
PB_MR1<P11MUX[14:12]> = “010”                       : Sets PB11 as SCK1.
PB_MR1<P12MUX[18:16]> = “010”                       : Sets PB12 as MOSI1.
PB_MR1<P13MUX[22:20]> = “010”                       : Sets PB13 as MISO1.
PB_CR<P10[21:20]> = “10”                            : Sets PB10(SS1) as an input pin.
PB_CR<P11[23:22]> = “10”                            : Sets PB11(SCK1) as an input pin.
PB_CR<P12[25:24]> = “10”                            : Sets PB12(MOSI1) as an input pin.
PB_CR<P13[27:26]> = “00”                            : Sets PB13(MISO1) as an output pin.
PB_PRCR<PUE10[21:20]> = “00”                        : Disables pull-up/pull-down at PB10 (SS1).
PB_PRCR<PUE11[23:22]> = “00”                        : Disables pull-up/pull-down at PB11 (SCK1).
PB_PRCR<PUE12[25:24]> = “00”                        : Disables pull-up/pull-down at PB12 (MOSI1).
PB_PRCR<PUE13[27:26]> = “00”                        : Disables pull-up/pull-down at PB13 (MISO1).

SPI1_CR<BITSZ[1:0]> = “00”                          : Sets the transmit-receive bit size to 8 bits.
SPI1_CR<MSBF[4]> = “1”                              : Selects MSB-first transmit.
SPI1_CR<MS[5]> = “0”                                : Selects SPI1 as a slave.
SPI1_CR<SSPOL[8]> = “1”                             : Sets the SS1 output signal high.
SPI1_CR<CPHA[3]> = “1”                              : Outputs data at each shift of an inverted phase signal.
```

```
SPI1_CR<CPOL[2]> = "1"               : Starts the clock for data sampling in the high state.
SPI1_BR<BR[7:0]> = "0100 1111"       : Sets the transmit-receive rate to 0.1 Mbps.
SPI1_EN<ENABLE[0]> = "1"             : Sets the transmit buffer to be initialized after the
SPI1_EN<ENABLE[0]> = "0"             first data transmit.
```

<Example 3> Data output configuration of master SPI0

```
SPI0_CR<SSOUT[12]> = "0"            : Sets master SPI0's SS output signal low.
SPI0_SR<TRDY[1]> READ               : Checks if the SPI0 transmit-receive buffer is ready for use.
PI0_EN<ENABLE[0]> = "0"             : Disables SPI0 to fill the SPI0 buffer with transmit data.

SPI0_TDR<TDR[16:0]> = "VALUE"       : Enters a transmit value to SPI0's transmit data register.

SPI0_En<ENABLE[0]> = "1"            : Enables SPI0 and starts outputting data.
SPI0_SR<TRDY[1]> READ               :  Checks  if  the  SPI0  transmit-receive  buffer  has  completed  data
                                    transmit.

SPIn_CR<SSOUT[12]> = "1"            : Sets master SPI0's SS output signal high.
SPI0_EN<ENABLE[0]> = "0"            : Disables SPI0.
```

<Example 4> Data output configuration of slave SPI1

```
SPI1_En<ENABLE[0]> = "1"            : Enables slave SPI1.
SPI1_SR<RRDY[0]> READ               : Checks if SPI1 holds receive data (0: no, 1: yes).
SPI1_RDR<RDR[16:0]> READ            : Reads SPI1's receive data register value.
```

13 Inter Integrated Circuit (I2C)

Inter Integrated Circuit (I2C) interface built in A33M11x series provides serial communications with internal and external devices via the I2C protocol.

Equipped with two channels, it supports both master and slave modes and capable of transmitting and receiving data in bytes through interrupts or polling. The I2C block is used to communicate with various peripherals that have the same bus type. When using the I2C capabilities of the A33M11x series, it is recommended to configure pins SCL and SDA as open-drain and then connect an external pull-up resistor to each of them to render their output signals “high.”

The I2C features the followings:

- Compliant with I2C protocol
  - Supports two channels
- Master and slave modes
- Multi-slave mode
  - 1:1 and N:N (up to 1008) slave devices
- Transfer rates configurable
  - Maximum transfer rate: 400KHz
- I2C interrupts
- 7-bit addressing
- Delay time can be set for pin SCL’s high or low waveform
- Hold time can be set for previous data
- Generates and detects STOP, START, and ACK signals

Table 54 introduces pins assigned for I2C interface.

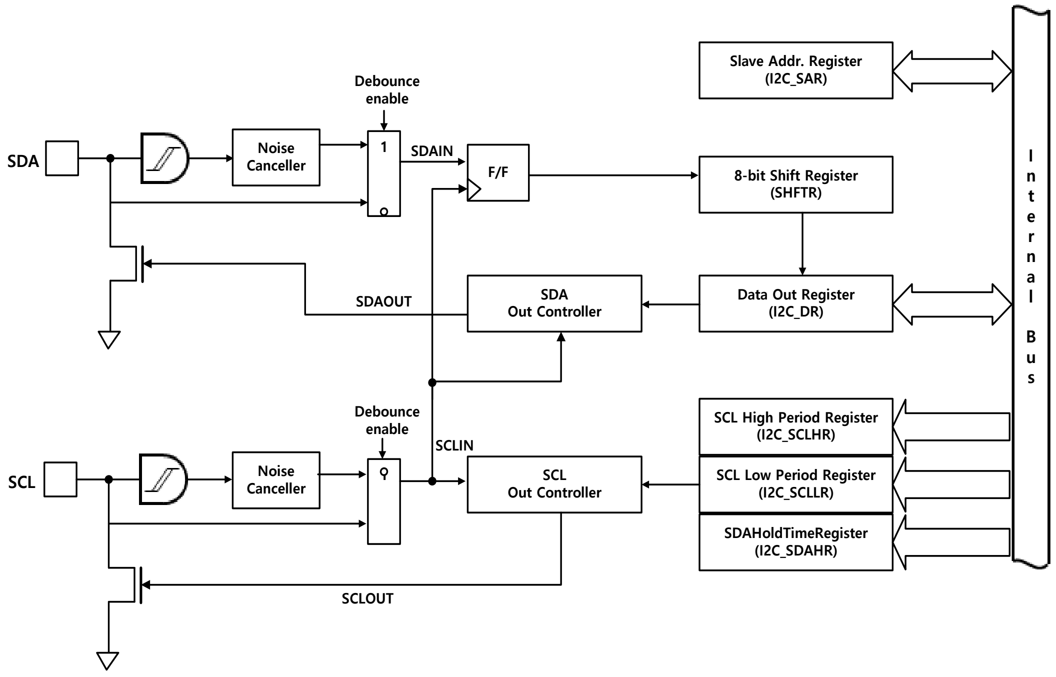
Table 54. Pin Assignment of I2C: External Pins

Pin name	Type	Description	Supported Packages	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
SCL0	I/O	I2C channel 0 serial clock bus line (open-drain)	O	O
SDA0	I/O	I2C channel 0 serial data bus line (open-drain)	O	O
SCL1	I/O	I2C channel 1 serial clock bus line (open-drain)	O	O
SDA1	I/O	I2C channel 1 serial data bus line (open-drain)	O	O

13.1 I2C block diagram

In this section, I2C interface block is described in a block diagram.

Figure 80. I2C Block Diagram



NOTE: 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.

13.2 Registers

Base address of I2C is introduced in the followings:

Table 55. Base Address of I2C Interface

Name	Base address
I2C0	0x4000_A000
I2C1	0x4000_A100

Table 56. I2C Register Map

Name	Offset	Type	Description	Reset value	Reference
I2Cn_DR	0x0000	RW	I^{2}C n data register	0x0000_00FF	13.2.1
I2Cn_SR	0x0008	RC	I^{2}C n status register	0x0000_0000	13.2.2
I2Cn_SAR	0x000C	RW	I^{2}C n slave address register	0x0000_0000	13.2.3
I2Cn_CR	0x0014	RW	I^{2}C n control register	0x0000_0000	13.2.4
I2Cn_SCLLR	0x0018	RW	I^{2}C n SCL low duration register	0x0000_FFFF	13.2.5
I2Cn_SCLHR	0x001C	RW	I^{2}C n SCL high duration register	0x0000_FFFF	13.2.6
I2Cn_SDAHR	0x0020	RW	I^{2}C n SDA hold register	0x0000_7FFF	13.2.7

NOTE: n = 0 and 1

13.2.1 I2Cn_DR: I2Cn data register

I2Cn_DR is a 32-bit register. This register stores the received byte-sized data or serial data for transmission.

I2C0_DR=0x4000_A000, I2C1_DR=0x4000_A100

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	ICDR
-	0xFF
-	RW

7 0	ICDR	In transmit mode, the data stored in this register is outputted from pin SDA in the serial data line. In receive mode, data received at pin SDA is stored in this register; hence, the stored data is loaded by reading this register.

13.2.2 I2Cn_SR: I2Cn status register

I2Cn_SR is a 32-bit R/W register. It displays the status of the I2C bus interface. Writing to the register clears the status bits. Once an I2C interrupt other than the stop interrupt occurs, the SCL line is set low. To release this SCL setting, an arbitrary value must be written to the SR register. This will clear the status of the TEND, STOP, SSEL, MLOST, and RXACK bits.

I2C0_SR=0x4000_A008, I2C1_SR=0x4000_A108

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			GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMOD	RXACK
-			0	0	0	0	0	0	0	0
-			RC	RC	RC	RC	RC	RC	RC	RC

7	GCALL	This bit signifies different things, depending on whether the I2C is in master or slave mode. (If the I2C module functions as a master, the bit represents whether or not the ACK address (AACK) has been received from a slave. If the I2C module is a slave, it indicates whether or not a general call has been detected.)	
		0	Master mode: AACK has not been received.
		1	Master mode: AACK has been received.
		0	Slave mode: A general call has not been detected.
		1	Slave mode: A general call has been detected.
6	TEND	One-byte transmit end flag	
		0	Transmit is in progress.
		1	Transmit has been completed.
5	STOP	Stop flag	
		0	No stop has been detected.
		1	A stop has been detected.
4	SSEL	Slave flag	
		0	The module has not been selected as a slave.
		1	The module has been selected as a slave.
3	MLOST	Mastership loss flag	
		0	Mastership has not been lost.
		1	Mastership has been lost.
2	BUSY	Bus busy flag	
		0	The I2C bus is idle.
		1	The I2C bus is busy.
1	TMOD	Transmit/receive mode flag	
		0	Receive mode
		1	Transmit mode
0	RXACK	Rx ACK flag	
		0	An Rx ACK has not been received.
		1	An Rx ACK has been received.

13.2.3 I2Cn_SAR: I2Cn slave address register

I2Cn_SAR is an 8-bit readable and writable register. The first seven bits store the address selected when the I2C module operates as a slave.

I2C0_SAR=0x4000_A00C, I2C1_SAR=0x4000_A10C

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			SVAD	GCEN
-			0x00	0
-			RW	RW

7 1	SVAD	7-bit slave address	
0	GCEN	Whether to enable or disable general call	
		0	Dismisses the general call address.
		1	Receives the general call address.

13.2.4 I2Cn_CR: I2Cn control register

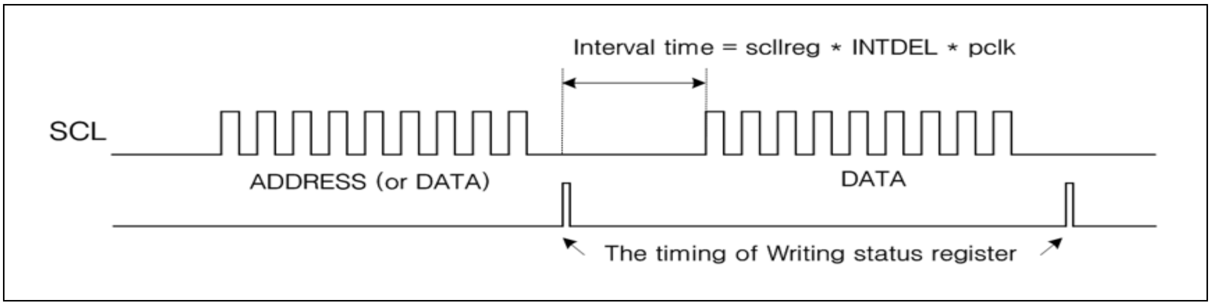
I2Cn_CR is a 32-bit R/W register. It sets the operating modes and enablement of the I2C module.

I2C0_CR=0x4000_A014, I2C1_CR=0x4000_A114

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	INTDEL	IIF	Reserved	SOFTRST	INTEN	ACKEN	Reserved	STOP	START
-	00	0	-	0	0	0	-	0	0
-	RW	RO	-	RW	RW	RW	-	RW	RW

9 8	INTDEL	Internal delay between address and data transfers (or between two data transfers)	
		0	1 * SCLL
		1	2 * SCLL
		2	4 * SCLL
		3	8 * SCLL
7	IIF	Interrupt flag	
		0	No interrupt has occurred or the flagged interrupt has been cleared.
		1	An interrupt has occurred.
5	SOFTRST	Whether to enable or disable the soft reset	
		0	Disables the soft reset.
		1	Enables the soft reset.
4	INTEN	Whether to enable or disable interrupts	
		0	Disables interrupts.
		1	Enables interrupts.
3	ACKEN	Whether to enable or disable ACK in receive mode	
		0	Does not receive an ACK signal after data reception.
		1	Receives an ACK signal after data reception.
1	STOP	Whether to enable or disable stop If this bit is set to 1 in transmit mode, the next transmission is stopped even if an ACK signal has been received.	
		0	Disables stop.
		1	Enables stop. Transmission is stopped when this bit is set to 1.
0	START	Transmission start in master mode	
		0	The I2C waits in slave mode.
		1	The I2C starts transmitting in master mode.

Figure 81. INTDEL (Inter Delay) in Master Mode


Interval time = scllreg * INTDEL * pclk
SCL
ADDRESS (or DATA)
DATA
The timing of Writing status register

13.2.5 I2Cn_SCLLR: I2Cn SCL low duration register

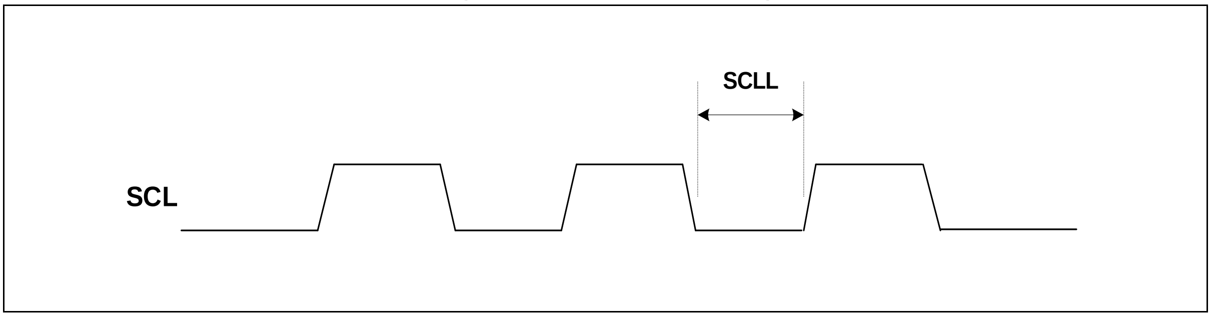
I2Cn_SCLLR is a 32-bit register. The SCL low duration time can be set in master mode.

I2C0_SCLLR=0x4000_A018, I2C1_SCLLR=0x4000_A118

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															
-																0xFFFF															
-																RW															

15 0	SCLL	SCL low duration value SCLL = (PCLK * SCLL[15:0]) + 2*PCLKs Default value: 0xFFFF.

Figure 82. SCL Low Timing



13.2.6 I2Cn_SCLHR: I2Cn SCL high duration register

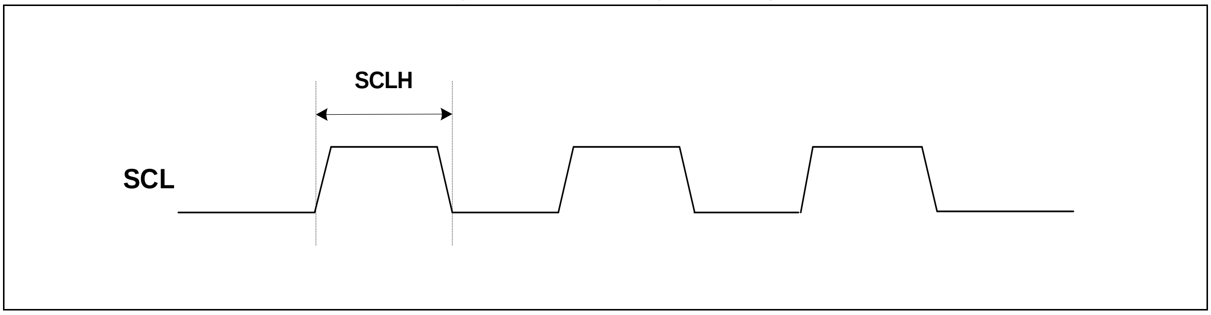
I2Cn_SCLHR is a 32-bit register. The SCL high duration time can be set in master mode.

I2C0_SCLHR=0x4000_A01C, I2C1_SCLHR=0x4000_A11C

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	
-		0xFFFF	
-		RW	

15 0	SCLH	SCL high duration value SCLH = (PCLK * SCLH[15:0]) + 3 PCLKs Default value: 0xFFFF.

Figure 83. SCL High Timing



13.2.7 I2Cn_SDAHR: I2Cn SDA hold register

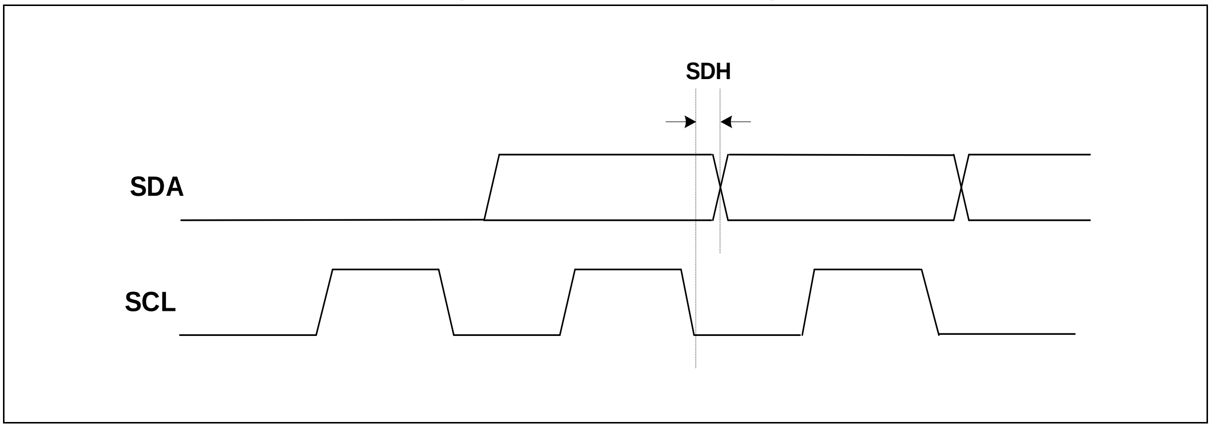
I2Cn_SDAHR is a 15-bit R/W register. The SCL hold time can be set in master mode.

I2C0_SDAHR=0x4000_A020, I2C1_SDAHR=0x4000_A120

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																	SDH														
-																	0x7FFF														
-																	RW														

14 0	SDH	SDA hold time value SDH = (PCLK * SDH[14:0]) + 4 PCLKs Default value: 0x7FFF.

Figure 84. SDA Hold Timing



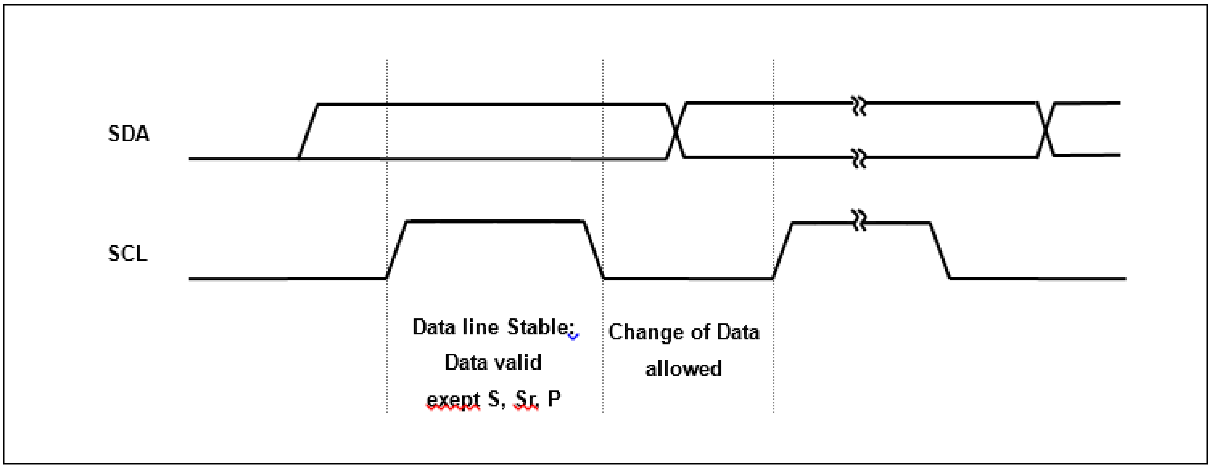
13.3 Functional description

13.3.1 I2C bit transfer

The data on the SDA line must be stable during the “H” period of the clock. The “H” or “L” state of the data line can only change when the clock signal on the SCL line is “L.” However, START (S), repeated START (Sr), and STOP (P) occur when the SDA data changes while the SCL signal is high.

The SDA line data must remain at “H” during the clock period. The “H” or “L” state of the data line can only change when the clock signal on the SCL line is “L”.

Figure 85. I2C Bus Bit Transfer: n = 0 and 1



13.3.2 START/repeated START/STOP

Within the procedure of the I2C-bus, unique situations arise which are defined as the START(S) and STOP(P) conditions (refer to Figure 86):

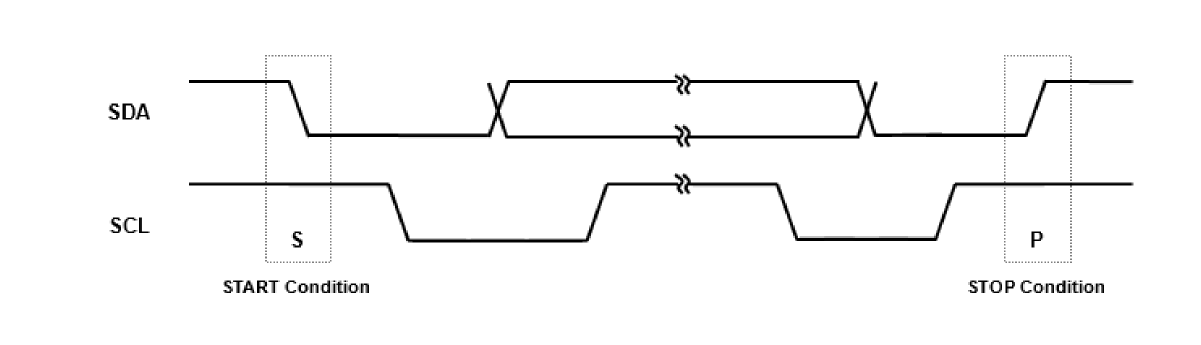
- An “H” to “L” transition on the SDA line while SCL is “H” is defined as the START(S) condition.
- An “L” to “H” transition on the SDA line while SCL is “H” is defined as the STOP condition.

These START and STOP conditions are always generated by the master. The bus is considered to be busy once the START condition occurs. And the bus is considered to be free again after an occurrence of the STOP condition.

Thus, the bus stays busy between a START and a STOP. If a repeated START(Sr) is generated instead of a STOP, the bus remains busy. In this respect, the START(S) and repeated START(Sr) conditions are functionally identical. For the remainder of this document, therefore, an S will be used to represent both the START and repeated START conditions, unless Sr is particularly relevant. The detection of the START and STOP conditions by devices connected to the bus is easy if the necessary interfacing hardware is incorporated.

However, microcontrollers with no such an interface have to sample the SDA line at least twice per clock period to sense the transition.

Figure 86. START and STOP Conditions: n = 0 and 1

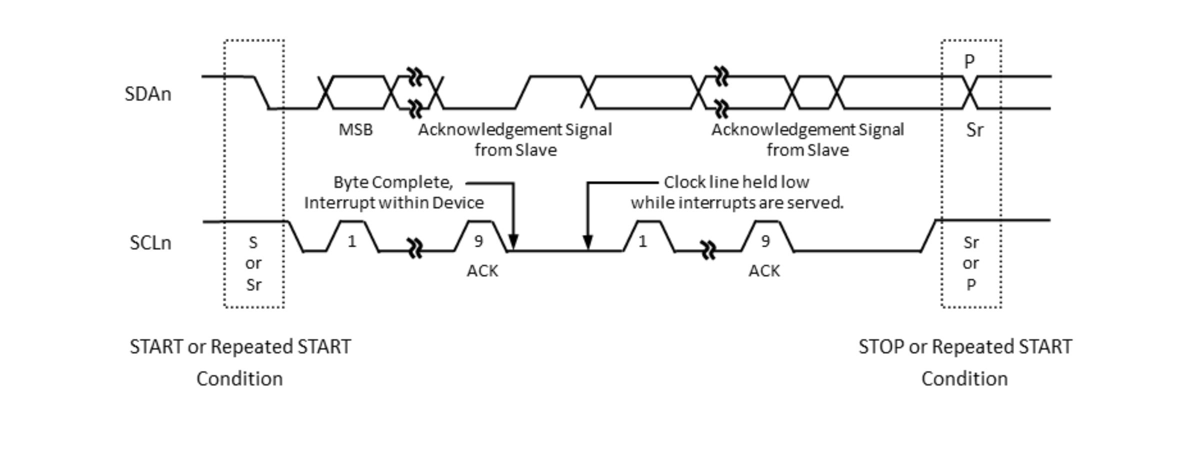


13.3.3 Data transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte must be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (refer to Figure 87). If a slave can’t receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL “L” 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 SCL.

A message starting with such an address can be terminated by an occurrence of the STOP condition, even during the transmission of a byte. In this case, no acknowledgement is generated.

Figure 87. I2C Bus Data Transfer: n = 0 and 1


SDAn
MSB
Acknowledgement Signal from Slave
Acknowledgement Signal from Slave
P
Sr
Byte Complete, Interrupt within Device
Clock line held low while interrupts are served.
SCLn
S or Sr
1
9
ACK
1
9
ACK
Sr or P
START or Repeated START Condition
STOP or Repeated START Condition

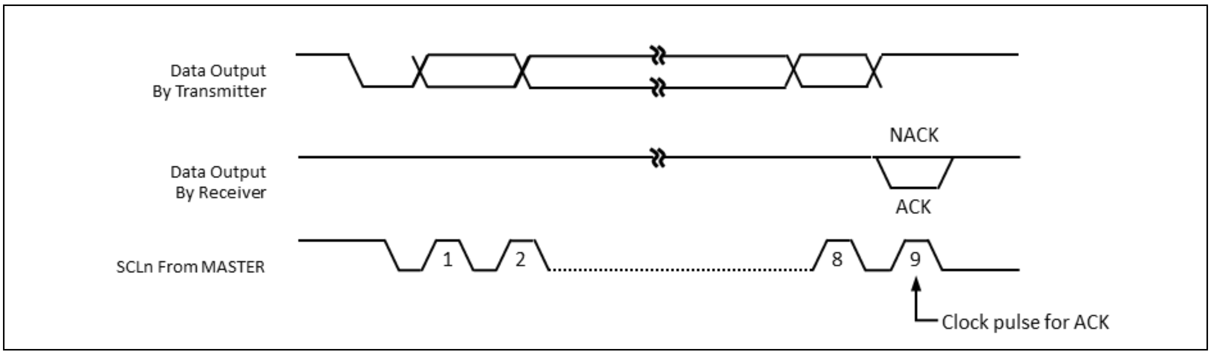
13.3.4 Acknowledge

A data transfer with acknowledgement is necessary. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line ("H") during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains at stable "L" during the "H" period of this clock pulse (refer to Figure 88).

Set-up and hold times must be considered too. When a slave doesn't acknowledge the slave address (for example, it's unable to receive or transmit because it's performing some real-time function), the data line must be left "H" by the slave. The master can then generate either the STOP condition to abort the transfer, or the repeated START condition to start a new transfer. If a slave-receiver acknowledges the slave address but can receive no more data bytes later during the transfer, the slave leaves the data line at "H" and the master generates either the STOP or the repeated START condition.

If a master-receiver is involved in a transfer, it must signal the slave-transmitter of the end of data by not generating 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 the STOP or repeated START condition.

Figure 88. I2C Bus Response: n = 0 and 1

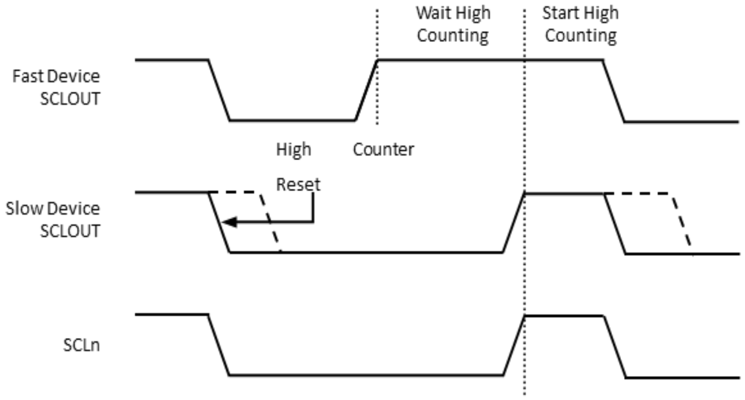


13.3.5 Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I2C-bus. Data is only valid during the “H” period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedure to take place.

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that an “H” to “L” transition on the SCL line will cause the devices concerned to start counting off their “L” period and, once a device clock has gone “L,” it will hold the SCL line in that state until the clock “H” state is reached.

Figure 89. Clock Synchronization during Arbitration: n = 0 and 1



NOTE: However, the “L” to “H” transition of this clock may not change the state of the SCL line if another clock is still within its “L” by the device with the longest “L” period. Devices with shorter “L” periods enter an “H” wait-state during this time.

When all devices concerned have counted off their “L” period, the clock line will be released and go “H.”

There will then be no difference between the device clocks and the state of the SCL line, and the devices will start counting their “H” periods. The first device to complete its “H” period will again pull the SCL line “L.”

13.3.6 Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate the START condition to the bus within the minimum hold time defined for this condition.

Arbitration takes place on the SDA line while the SCL line is in the "H" level in the condition that one master transmits the "H" level but another master is transmitting the "L" level; as a result, the master transmitting the "H" level switches off its DATA output stage because the level on the bus doesn't correspond to its own level.

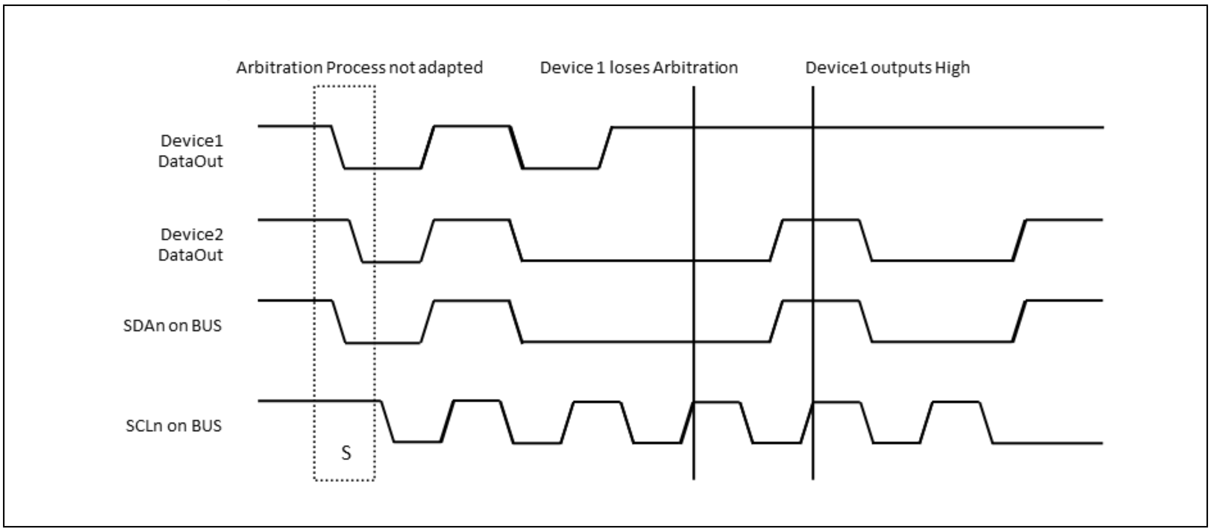
Arbitration can continue for many bits. Its first stage is the comparison of the address bits. If the masters are trying to address the same device, arbitration proceeds with comparing either the data-bits (if they are master-transmitters) or the acknowledge-bits (if they are master-receivers).

Because address and data information on the I2C bus is determined by the winning master, no information is lost during the arbitration process. A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master also incorporates a slave function and it loses arbitration during the addressing stage, it is possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode.

Figure 90 shows the arbitration procedure for two masters. Of course, more may be involved (depending on how many masters are connected to the bus). As soon as a difference is made between the internal data level of the master generating Device1 DataOut and the actual level on the SDA line, its data output is switched off, which means that an "H" output level is then connected to the bus. This will not affect the data transfer initiated by the winning master.

Figure 90. Arbitration Process between Two Masters: n = 0 and 1



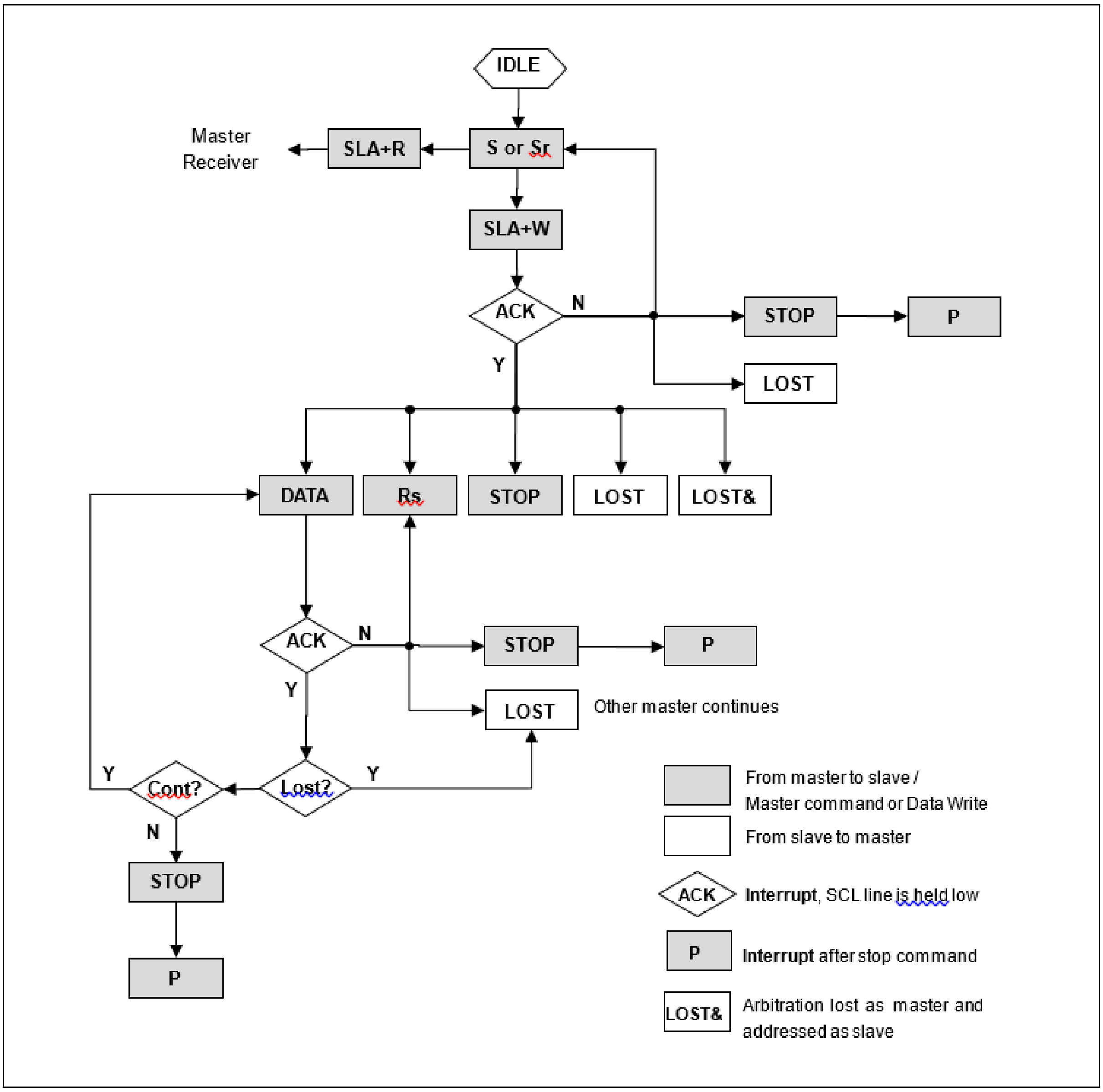
13.3.7 I2C operation

The I2C module uses interrupts. Once an interrupt is serviced, the CR register's 7th bit is flagged. The SR register shows I2C bus status information; the SCL line stays "L" until a certain value is written to the register. The status register can be cleared by being written to.

13.3.7.1 Master transmitter

Figure 91 shows a flowchart of the transmitter in master mode.

Figure 91. Master Transmitter Flowchart


IDLE
Master Receiver
SLA+R
S or Sr
SLA+W
ACK
N
Y
STOP
P
LOST
DATA
Rs
STOP
LOST
LOST&
ACK
N
Y
STOP
P
LOST
Other master continues
Y
Cont?
Lost?
Y
N
STOP
P
From master to slave / Master command or Data Write
From slave to master
ACK
Interrupt, SCL line is held low
P
Interrupt after stop command
LOST&
Arbitration lost as master and addressed as slave

To operate the I2C in master transmitter mode, you must configure it in the following steps:

1. Set the SCU_PER2 and SCU_PCER2 registers' I2Cn bits to enable and clock the I2C.
2. Set the I2Cn_CR register's INTEN bit to enable the I2C interrupts.
3. Enter "SLA+W" in the I2Cn_DR register ("SLA" = slave address, "W" = write). "W" must be set to 0 in master transmitter mode. Note that I2Cn_DR register is used for both an address and data.
4. Set the I2Cn_SCLLR and I2Cn_SCLHR registers to configure the SCL transfer speed.
5. Set the I2Cn_SDAHR register, which determines the time SCL maintains the SDA value during the "L" period.
6. Write a 1 to I2Cn_CR's START bit to transmit the START condition. And then, configure interrupts and ACK processing conditions. Once the START bit is set enabled, the 8-bit data in I2Cn_DR is transmitted at the specified transfer speed.
7. Next is the ACK processing sector for the data transmitted by the master. The slave receives a 7-bit address and a 1-bit transmit-receive direction. The master checks for an ACK response in the "H" sector of the ninth clock pulse. If the master has bus priority, the GCALL interrupt occurs, regardless of whether or not an ACK has been received. If the I2C loses its bus mastership, I2Cn_SR's MLOST bit is set to 1. And the I2C either waits in idle mode or is operated in slave mode. To operate the I2C in slave mode when it has lost its bus mastership, I2Cn_CR's ACKEN bit must be set to 1 and I2Cn_SAR's SVAD bit field must be set to the specified slave address. When the I2C operates in slave transmit or receive mode, its SCL must be maintained in the "L" level, so that the I2C can decide whether to continue or stop transmission. The following is an example for how to operate the I2C when it does not lose its mastership during the first byte transmission.

   The I2C (master) performs one of the following operations regardless of whether or not it has received an ACK signal from the slave:

   Case 1. The master receives an ACK signal from the slave. Thereby, the master sends data to the slave. In this case, the data is written to the I2Cn_DR register for transmission.

   Case 2. Even if an ACK signal has been received, the master can stop transmission. In this case, I2Cn_CR's STOP bit must be set enabled.

   Case 3. The master sends the consecutive START condition without checking for an ACK signal. In this case, "SLA+R/W" must be written to I2Cn_DR; and I2Cn_CR's START bit must be set to 1.

The SCL line is released when one of the above is performed and an arbitrary value (0xFF) is written to the I2Cn_SR register. For the Case 1, go to step 7; For the Case 2, go to step 9; For the Case 3, transmit the data in the I2Cn_DR register and go to step 6. If the transfer direction bit is set to 1, go to the section on master receiver.

8. One byte of data is transmitted. Bus arbitration is valid during data transmission.

9. The ACK processing sector for the address packet sent from the master. The I2C maintains the SCL in the “L” level. If the I2C’s bus mastership is taken over by another master during data transmission arbitration, I2Cn_SR’s MLOST bit is set to 1. In this case, the I2C waits until it becomes idle. Once the data in I2Cn_DR is completely transmitted, the I2C triggers the TEND interrupt.

   The I2C (master) performs one of the following operations regardless of whether or not it has received an ACK signal from the slave:

   Case A. The master receives an ACK signal from the slave. Thereby, the master sends data to the slave. In this case, the data is written to the I2Cn_DR register for transmission.

   Case B. Even if an ACK signal has been received, the master can stop transmission. In this case, I2Cn_CR’s STOP bit must be set enabled.

   Case C. The master sends the consecutive START condition without checking for an ACK signal. In this case, “SLA+R/W” must be written to I2Cn_DR; and I2Cn_CR’s START bit must be set to 1.

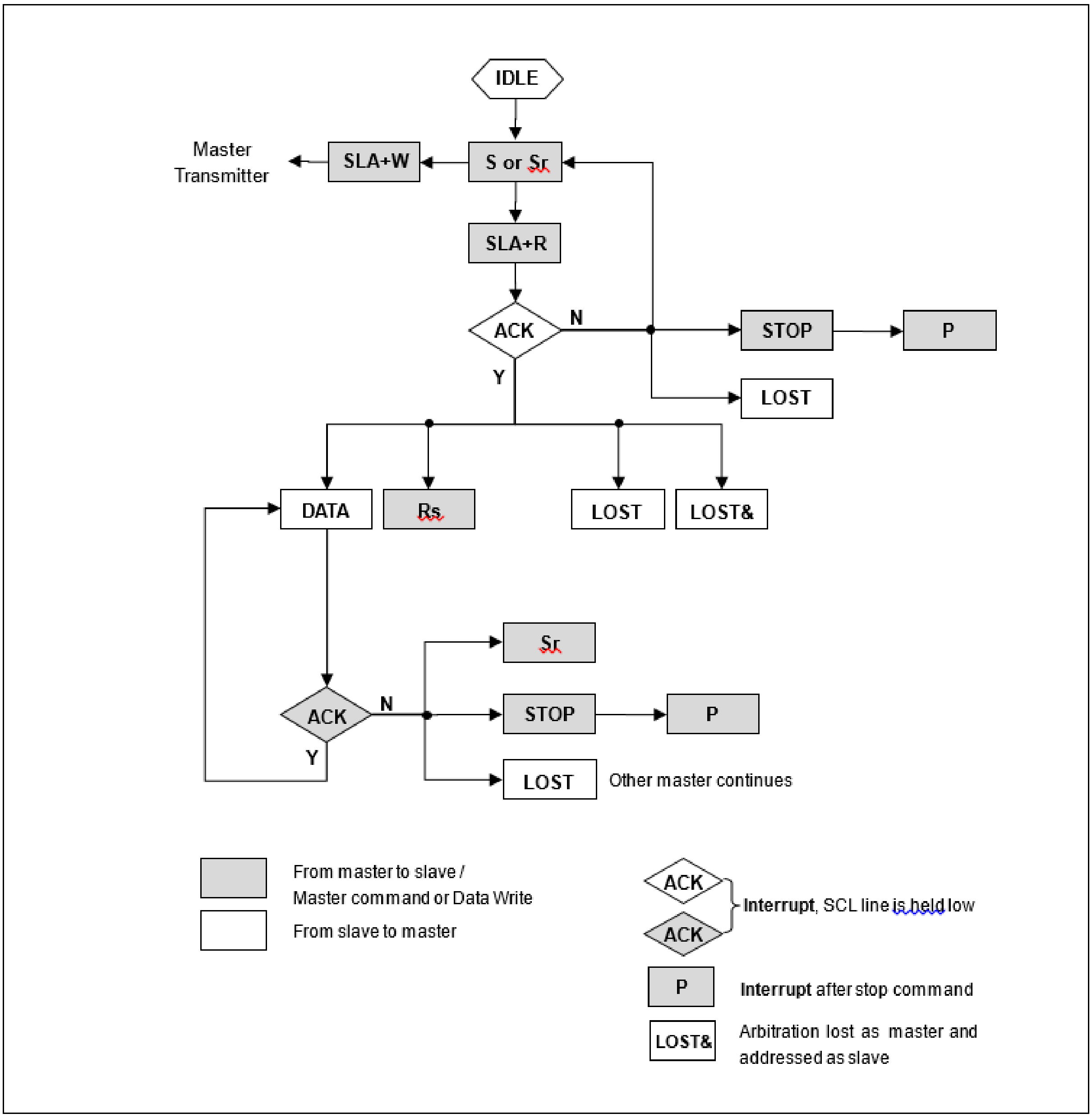
   The SCL line is released when one of the above is performed and an arbitrary value (0xFF) is written to the I2Cn_SR register. For the Case A, go to step 7; For the Case B, go to step 9; For the Case C, transmit the data in the I2Cn_DR register and go to step 6. If the transfer direction bit is set to 1, go to the section on master receiver.

10. The last step is to process the STOP interrupt. This interrupt indicates that data transfer between a master and slave has been stopped by the STOP bit. To clear the I2Cn_SR register, write an arbitrary value (0xFF) to the register. This causes the I2C to become idle.

13.3.7.2 Master receiver

Figure 92 shows a flowchart of the receiver in master mode.

Figure 92. Master Receiver Flowchart



To operate the I2C in master receiver mode, you must configure it in the following steps:

1. Set the SCU_PER2 and SCU_PCER2 registers' I2Cn bits to enable and clock the I2C.
2. Set the I2Cn_CR register's INTEN bit to enable the I2C interrupts.

3. Enter “SLA+R” in the I2Cn_DR register (“SLA” = slave address, “R” = read). “R” must be set to 1 in master transmitter mode. Note that I2Cn_DR register is used for both an address and data.

4. Set the I2Cn_SCLLR and I2Cn_SCLHR registers to configure the SCL transfer speed.

5. Set the I2Cn_SDAHR register, which determines the time SCL maintains the SDA value during the “L” period.

6. Write a 1 to I2Cn_CR’s START bit to transmit the START condition. And then, configure interrupts and ACK processing conditions. Once the START bit is set enabled, the 8-bit data in I2Cn_DR is transmitted at the specified transfer speed.

7. The ACK processing sector for the address packet sent from the master. The slave receives a 7-bit address and a 1-bit transmit-receive direction. The master checks for an ACK response in the “H” sector of the ninth clock pulse. If the master has bus priority, the GCALL interrupt occurs, regardless of whether or not an ACK has been received. If the I2C loses its bus mastership, I2Cn_SR’s MLOST bit is set to 1. And the I2C either waits in idle mode or is operated in slave mode. To operate the I2C in slave mode when it has lost its bus mastership, I2Cn_CR’s ACKEN bit must be set to 1 and I2Cn_SAR’s SVAD bit field must be set to the specified slave address. When the I2C operates in slave transmit or receive mode, its SCL must be maintained in the “L” level, so that the I2C can decide whether to continue or stop transmission. The following is an example for how to operate the I2C when it does not lose its mastership during the first byte transmission.

   The I2C (master) performs one of the following operations regardless of whether or not it has received an ACK signal from the slave:

   Case 1. The master receives an ACK signal from the slave. Thereby, the master sends data to the slave. In this case, the data is written to the I2Cn_DR register for transmission.

   Case 2. Even if an ACK signal has been received, the master can stop transmission. In this case, I2Cn_CR’s STOP bit must be set enabled.

   Case 3. The master sends the consecutive START condition without checking for an ACK signal. In this case, “SLA+R/W” must be written to I2Cn_DR; and I2Cn_CR’s START bit must be set to 1.

   The SCL line is released when one of the above is performed and an arbitrary value (0xFF) is written to the I2Cn_SR register. For the Case 1, go to step 7; For the Case 2, go to step 9; For the Case 3, transmit the data in the I2Cn_DR register and go to step 6. If the transfer direction bit is set to 1, go to the section on master receiver.

8. One byte of data is received.

9. The ACK processing sector for the address packet received from the slave. The I2C maintains the SCL in the “L” level. Once a byte of data is received, TEND interrupt occurs in the I2C.

   The I2C (master) performs one of the following operations regardless of the RXACK flag in I2Cn_SR:

   Case A. The master continues to receive data from the slave. The I2Cn_CR register’s ACKEN bit is set to 1 to transmit an ACK signal so that the slave can transmit the next data.

   Case B. The master receiver stops receiving because an ACK is not generated in the next receive data. To do so, I2Cn_CR’s ACKEN bit must be cleared.

   Case C. When an ACK signal is not detected, the master stops data transfer. In this case, I2Cn_CR’s STOP bit must be set enabled.

   Case D. ACK signals are not detected and the master sends the consecutive START condition. In this case, “SLA+R/W” must be loaded to the I2Cn_DR register; and I2Cn_CR’s START bit must be set to 1.

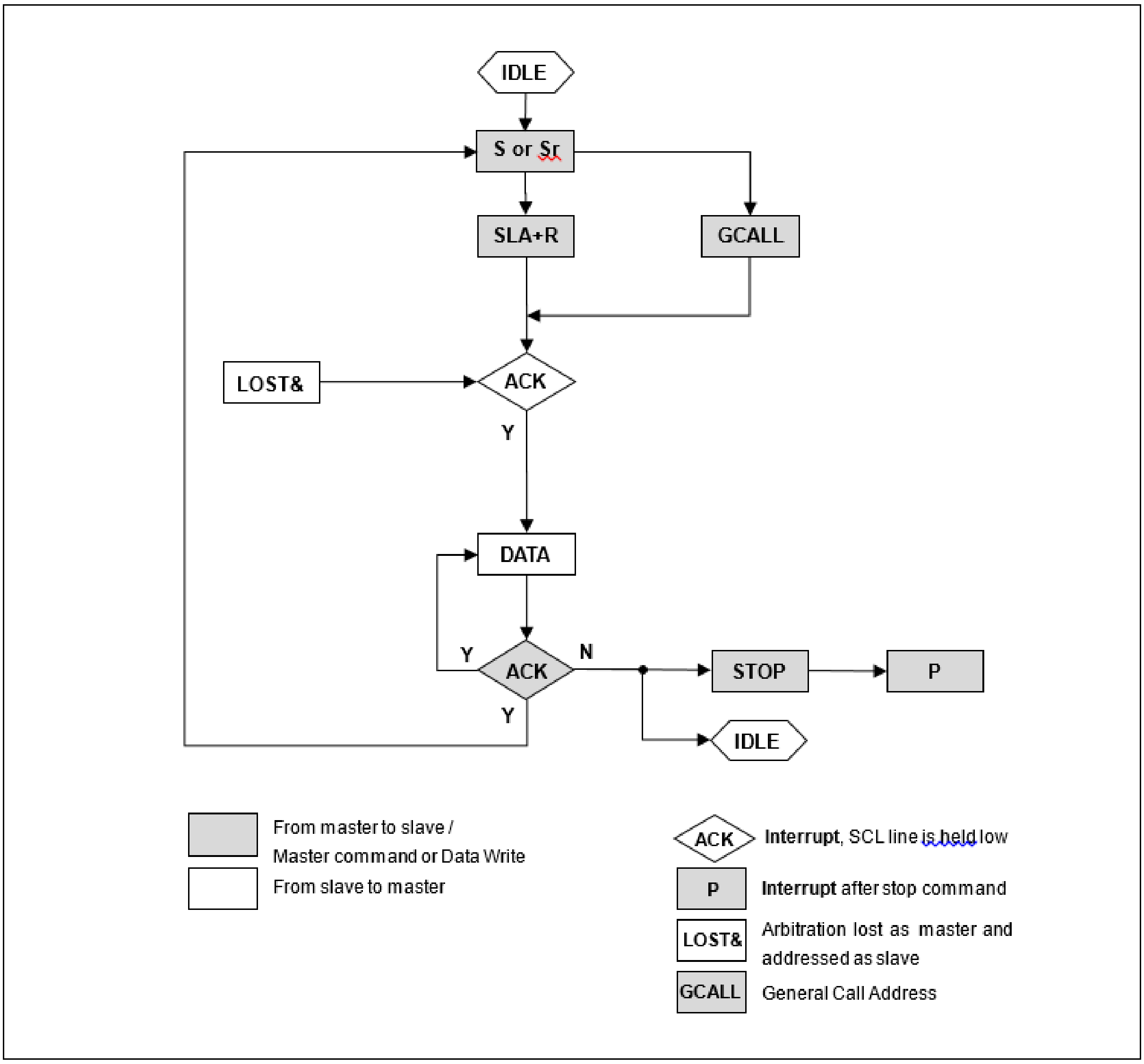
   The SCL line is released when one of the above is performed and an arbitrary value (0xFF) is written to the I2Cn_SR register. For the Cases A and B, go to step 7; For the Case C, go to step 9; For the Case D, transmit the data in the I2Cn_DR register and go to step 6. If the transfer direction bit is set to 0, go to the section on master transmitter.

10. The last step is to process the STOP interrupt. This interrupt indicates that data transfer between a master and slave has been stopped by the STOP bit. To clear the I2Cn_SR register, write an arbitrary value (0xFF) to the register. This causes the I2C to become idle.

13.3.7.3 Slave transmitter

Figure 93 shows a flowchart of the transmitter in slave mode.

Figure 93. Slave Transmitter Flowchart


IDLE
S or Sr
SLA+R
GCALL
LOST&
ACK
Y
DATA
Y
ACK
N
Y
STOP
P
IDLE
From master to slave /
Master command or Data Write
From slave to master
ACK Interrupt, SCL line is held low
P Interrupt after stop command
LOST& Arbitration lost as master and addressed as slave
GCALL General Call Address

To operate the I2C in slave transmitter mode, you must configure it in the following steps:

1. If the main operating clock (PCLK) is slower than the SCL, you must set I2Cn_SDAHR to “0x0000,” so that the SDA is changed at the falling edge of the SCL. The SDA hold time is obtained by multiplying the ICn_SDAHR register value by the PCLK period. If the SDA hold time is longer than the SCL period, data cannot be properly transmitted.
2. Set the SCU_PER2 and SCU_PCER2 registers’ I2Cn bits to enable and clock the I2C.
3. Set the I2Cn_CR register’s INTEN bit to enable the I2C interrupts.

4. When the START condition is detected, the I2C receives one byte of data and compares it with the SVAD bit field in the I2Cn_SAR register. In the case that I2Cn_SAR's GCALLEN bit is set enabled, the I2C raises the "general call" flag if the received data is "0x00."

5. If the received address does not match I2Cn_SAR's SVAD bit field, the I2C becomes idle and waits until another START bit is detected. If the received address matches the address in the SVAD bit field, the ACKEN bit becomes enabled and the I2C triggers the SSEL interrupt and maintains the SCL line in the "L" level. If the ACKEN bit is set disabled, the I2C becomes idle even though the received address matches the SVAD bit field. When the SSEL interrupt occurs, you must load the data in I2Cn_DR for transmission and release the SCL line by writing an arbitrary value to I2Cn_SR.

6. One byte of data is transmitted.

7. Once the transmission is completed, the I2C triggers the TEND interrupt regardless of an ACK signal from the master, and it maintains the SCL line in the "L" level. The slave performs one of the following operations:

   Case 1. If a NACK signal is detected, the I2C waits for the STOP condition or consecutive START condition.

   Case 2. If an ACK signal from the master is detected, the I2C loads the next data in the I2Cn_DR register.

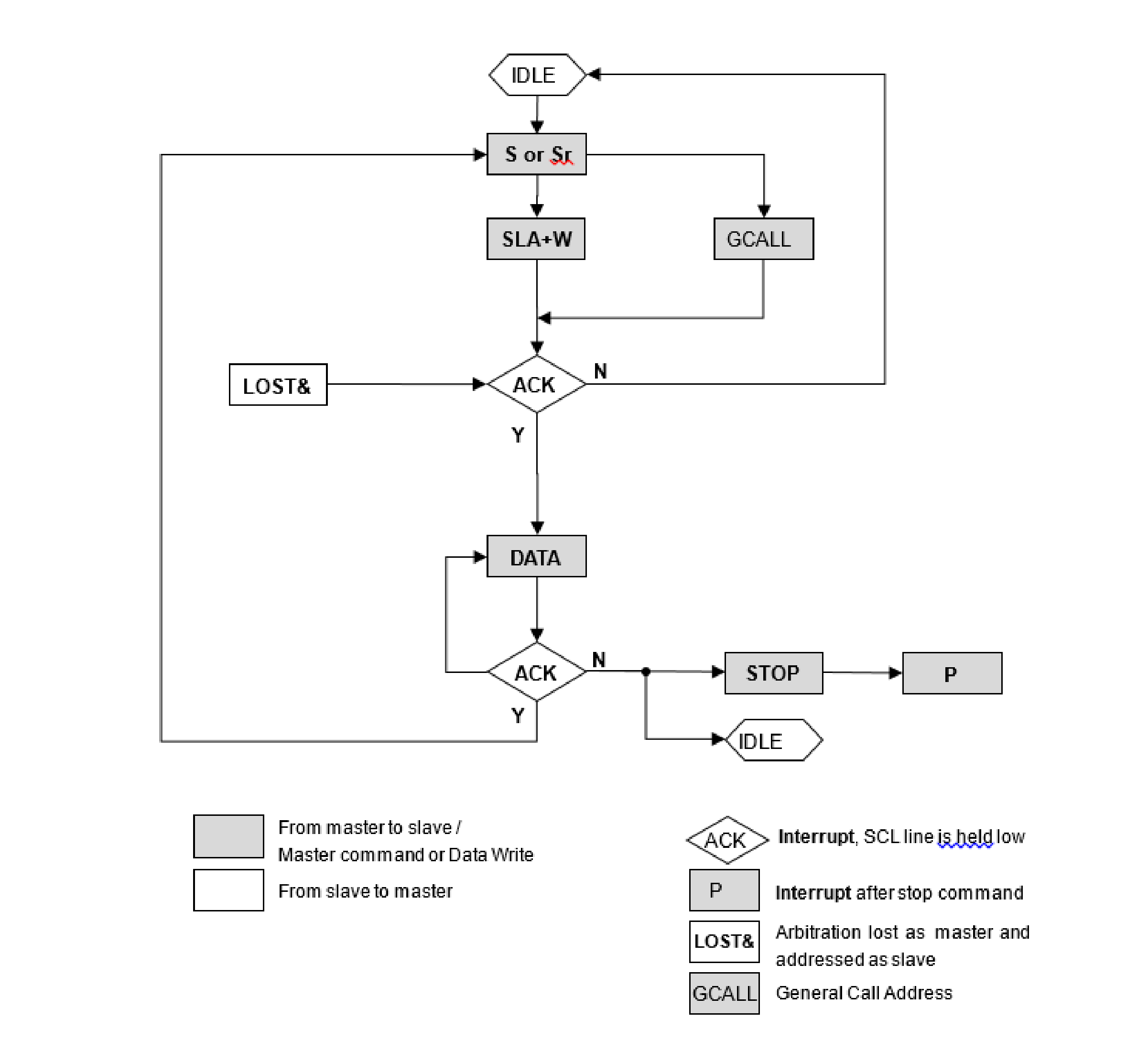
   Once one of the above operations is performed, release the SCL line by writing an arbitrary value to I2Cn_SR. For the Case 1, go to step 7 and end the communication. For the Case 2, go to step 5. In either case, the consecutive START condition can be detected; then, go to step 4.

8. The last step is to process the STOP interrupt. This interrupt indicates that data transfer between a master and slave has been completed. To clear the I2Cn_SR register, write an arbitrary value (0xFF) to the register. This causes the I2C to become idle.

13.3.7.4 Slave receiver

Figure 94 shows a flowchart of the Receiver in slave mode.

Figure 94. Slave Receiver Flowchart


IDLE
S or Sr
SLA+W
GCALL
LOST&
ACK
N
Y
DATA
ACK
N
Y
STOP
P
IDLE
From master to slave / Master command or Data Write
From slave to master
ACK Interrupt, SCL line is held low
P Interrupt after stop command
LOST& Arbitration lost as master and addressed as slave
GCALL General Call Address

To operate the I2C in slave receiver mode, you must configure it in the following steps:

1. If the main operating clock (PCLK) is slower than the SCL, you must set I2Cn_SDAHR to "0x0000," so that the SDA is changed at the falling edge of the SCL. The SDA hold time is obtained by multiplying the ICn_SDAHR register value by the PCLK period. If the SDA hold time is longer than the SCL period, data cannot be properly transmitted.

2. Set the SCU_PER2 and SCU_PCER2 registers' I2Cn bits to enable and clock the I2C.

3. Set the I2Cn_CR register’s INTEN bit to enable the I2C interrupts.

4. When the START condition is detected, the I2C receives one byte of data and compares it with the SVAD bit field in the I2Cn_SAR register. In the case that I2Cn_SAR’s GCALLEN bit is set enabled, the I2C raises the “general call” flag if the received data is “0x00.”

5. If the received address does not match I2Cn_SAR’s SVAD bit field, the I2C becomes idle and waits until another START bit is detected. If the received address matches the address in the SVAD bit field, the ACKEN bit becomes enabled and the I2C triggers the SSEL interrupt and maintains the SCL line in the “L” level. If the ACKEN bit is set disabled, the I2C becomes idle even though the received address matches the SVAD bit field. If the SSEL interrupt occurs, the I2C is ready to receive data. Release the SCL line by writing an arbitrary value to I2Cn_SR.

6. One byte of data is received.

7. Once the reception is completed, the I2C triggers the TEND interrupt regardless of an ACK signal from the master, and it maintains the SCL line in the “L” level. The slave performs one of the following operations:

   Case 1. If a NACK signal is detected (ACKEN = 0), the I2C waits for the STOP condition or consecutive START condition.

   Case 2. If an ACK signal from the master is detected (ACKEN = 1), the I2C can continue receiving data from the master.

   Once one of the above operations is performed, release the SCL line by writing an arbitrary value to I2Cn_SR. For the Case 1, go to step 7 and end the communication. For the Case 2, go to step 5. In either case, the consecutive START condition can be detected; then, go to step 4.

8. The last step is to process the STOP interrupt. This interrupt indicates that data transfer between a master and slave has been completed. To clear the I2Cn_SR register, write an arbitrary value (0xFF) to the register. This causes the I2C to become idle.

13.3.8 Setting examples

<Example 1> Initial configuration of master I2C0

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                : Unlocks the SCU registers.
SCU_PER2<I2C0[4]> = “1”                         : Enables the I2C0 peripheral.
SCU_PCER2<I2C0[4]> = “1”                        : Enables the I2C0 peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”                : Enables PORTEN (enter 0x15 and then 0x51).
PCU_PORTEN<PORTEN[7:0] = “0x51”

PC_MR2<P8MUX[2:0]> = “01”                       : Sets Port C pin 8 as SDA0.
PC_MR1<P7MUX[30:28]> = “01”                     : Sets Port C pin 7 as SCL0.
PC_CR<P8[17:16]> = “01”                         : Sets Port C pin 8 for SDA0 open-drain output.
PC_CR<P7[15:14]> = “01”                         : Sets Port C pin 7 for SCL0 open-drain output.
PC_PRCR<PUE8[17:16]> = “00”                     : Disables SDA0 pull-up/pull-down at Port C pin 8.
PC_PRCR<PUE7[15:14]> = “00”                     : Disables SCL0 pull-up/pull-down at Port C pin 7.

I2C0_CR<SOFTRESET[5]> = “1”                     : Initializes the I2C0 serial device’s internal register.
I2C0_CR<INTERVAL[9:8]> = “01”                   :  Sets  the  I2C0  internal  delay  option  value  (Delay  =
                                                I2Cn_SCLLR*2).
I2C0_CR<IINTEN[4]> = “1”                        : Enables the I2C0 interrupt.
I2C0_SCLL<SCL[31:0]> = ”0”                      : Initializes the I2C SCL low duration register.
I2C0_SCLH<SCLH[31:0]> = “0”                     : Initializes the I2C SCL high duration register.
I2C0_SDH<SDH[31:0]> = “0”                       : Initializes the I2C SDA hold register.
I2C0_CR<ACKEN[3]> = “1”                         : Sets I2C0 to generate an acknowledge signal after receiving
                                                data.

I2C0_SAR<SVAD[7:1]> = “010 0000”                : Enters the 7-bit slave address “0x20.”
```

14 Motor Pulse Width Modulation (MPWM)

Motor Pulse Width Modulation (MPWM) modules are programmable motor controllers optimized for three-phase AC and DC motor control applications.

Each MPWM module is equipped with three channels that can generate their respective pairs of outputs. The MPWM counter is clocked from the SCU block. As the clock determines MPWM resolution and period, you must select an appropriate MPWM clock before enabling the MPWM module.

The MPWM Normal Mode of A33M11x series features the followings:

- 16-bit counter
- Six output channels for motor control
- Dead-time rising or falling area
- Handling of protection and overvoltage events
- Six ADC trigger sources
- Interval interrupt mode (Only a period interrupt is used)
- Up-count and down-count modes

The MPWM Individual Mode of A33M11x series features the followings. (The MPWM module supports an Individual mode to enable various applications, such as IH cookers.):

- 16-bit counter
- Different periods and dead times (rising/falling) configurable for phases U, V, and W
- Handling of different protection and overvoltage events for phases U, V, and W
- Different interrupts for phases U, V, and W (except for the protection and overvoltage interrupts)
- Different protection and overvoltage events for phases U, V, and W (e.g., if protection occurs for phase V, only the phase V output becomes inactive while phases U and W remain operational).
- Capture functionality

Table 57 introduces pins assigned for MPWM.

Table 57. Pin Assignment of MPWM: External Pins

Pin name	**Type**	**Description**	**Supported Packages**	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
MP0UH/L MP0VH/L MP0WH/L	O	MPWM0 H/L side output ports of phases U, V, and W	O	O
MP1UH/L MP1VH/L MP1WH/L	O	MPWM1 H/L side output ports of phases U, V, and W	O	O
PRTIN0U OVIN0U	I	MPWM0 protection and overvoltage input pins dedicated to MPWM0 phase U in Individual mode	O	O
PRTIN1U OVIN1U	I	MPWM1 protection and overvoltage input pins dedicated to MPWM1 phase U in Individual mode	O	O
PRTINEV OVINEV	I	Protection and overvoltage input pins dedicated to MPWM0/1 phase V in Individual mode	O	O
PRTINEW OVINEW	I	Protection and overvoltage input pins dedicated to MPWM0/1 phase W in Individual mode	O	X
CAPEU CAPEV CAPEW	I	Input pins in MPWM0/1 dedicated to capturing in Individual mode	O	O
SCAPEU SCAPEV SCAPEW	I	Input pins in MPWM0/1 dedicated to sub-capturing in Individual mode	O	O
Normal mode	MPWM0 Protection		MPWM1 Protection	
	PRTIN0U, OVIN0U		PRTIN1U, OVIN1U	
Individual mode	U	PRTIN0U, OVIN0U	PRTIN1U, OVIN1U	
	V	PRTINEV	OVINEV	
	W	PRTINEW	OVINEW	

NOTE: In individual PWM mode, V and W of PRTINE/ OVINE operate simultaneously with MPWM0 and MPWM1.

14.1 MPWM block diagram

Figure 95 describes normal mode of MPWM in block diagram.

Figure 95. MPWM Block Diagram: Normal Mode

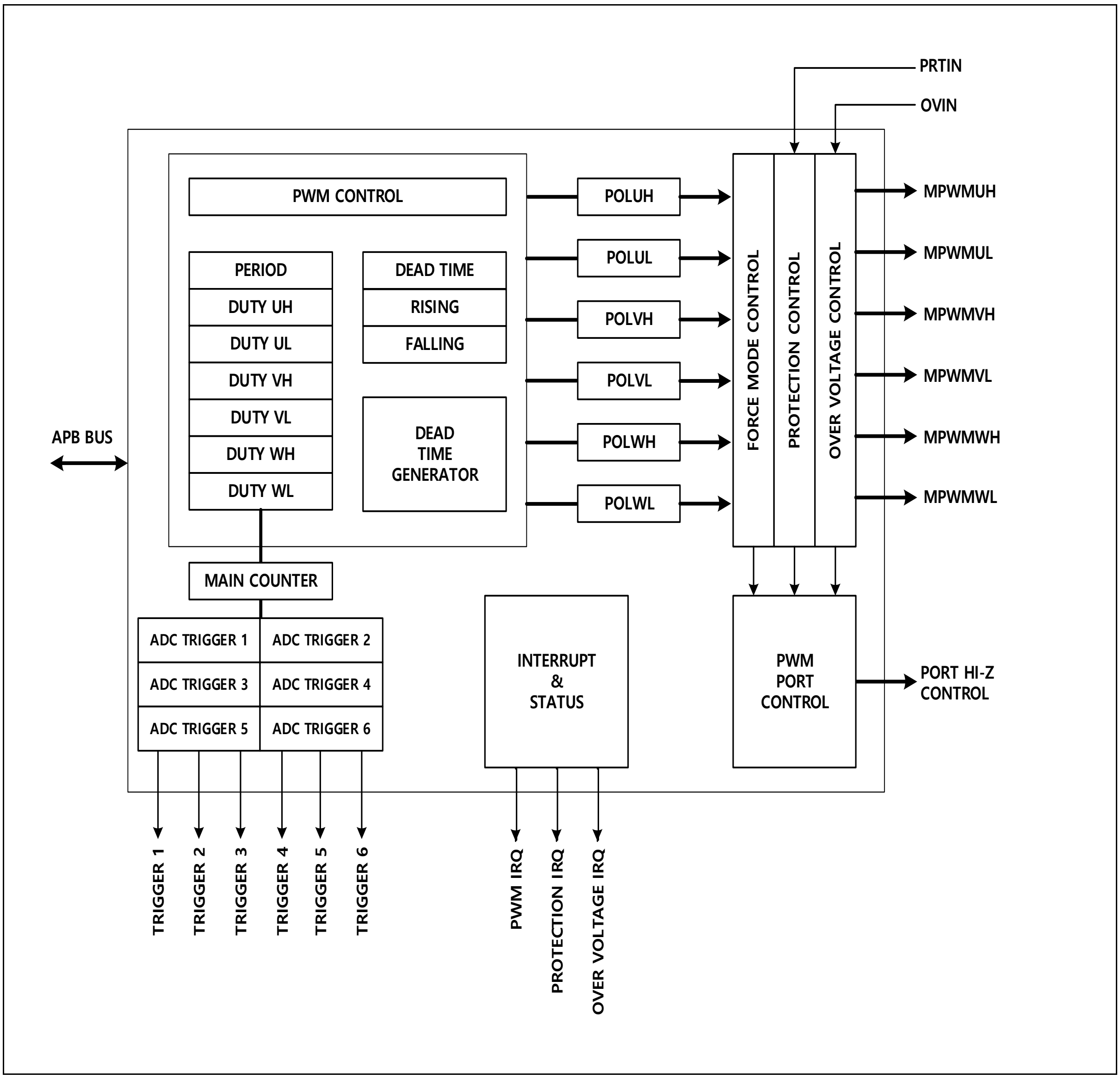


Figure 96 describes individual mode of MPWM in block diagram.

Figure 96. MPWM Block Diagram: Individual Mode

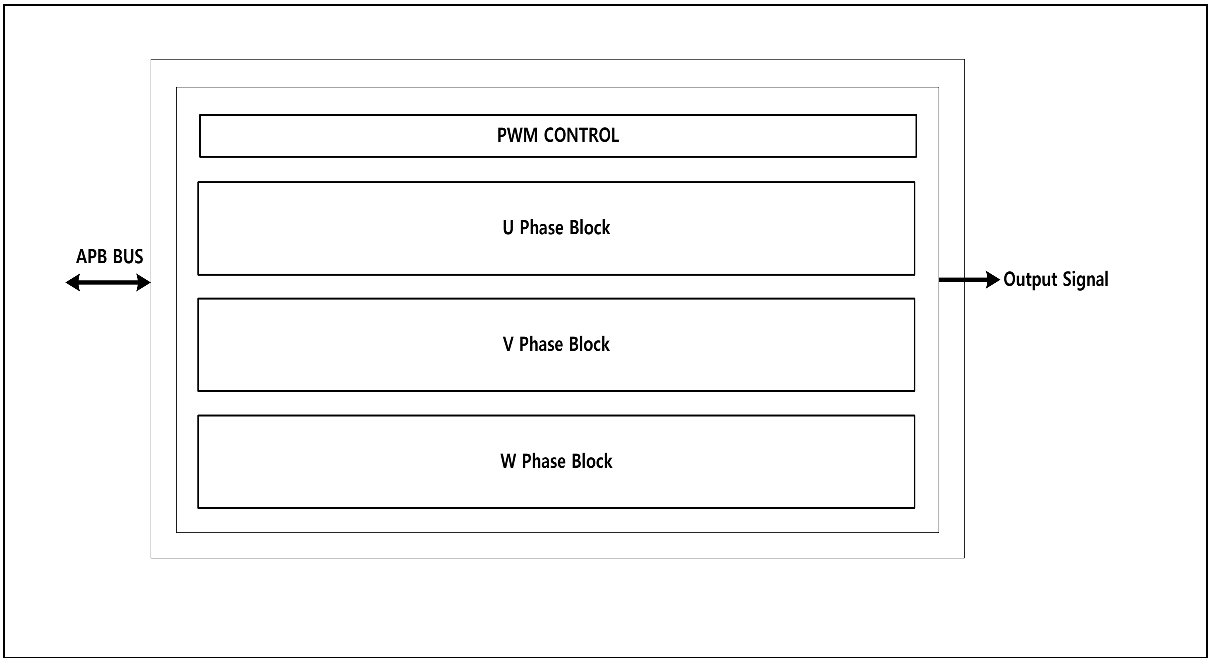
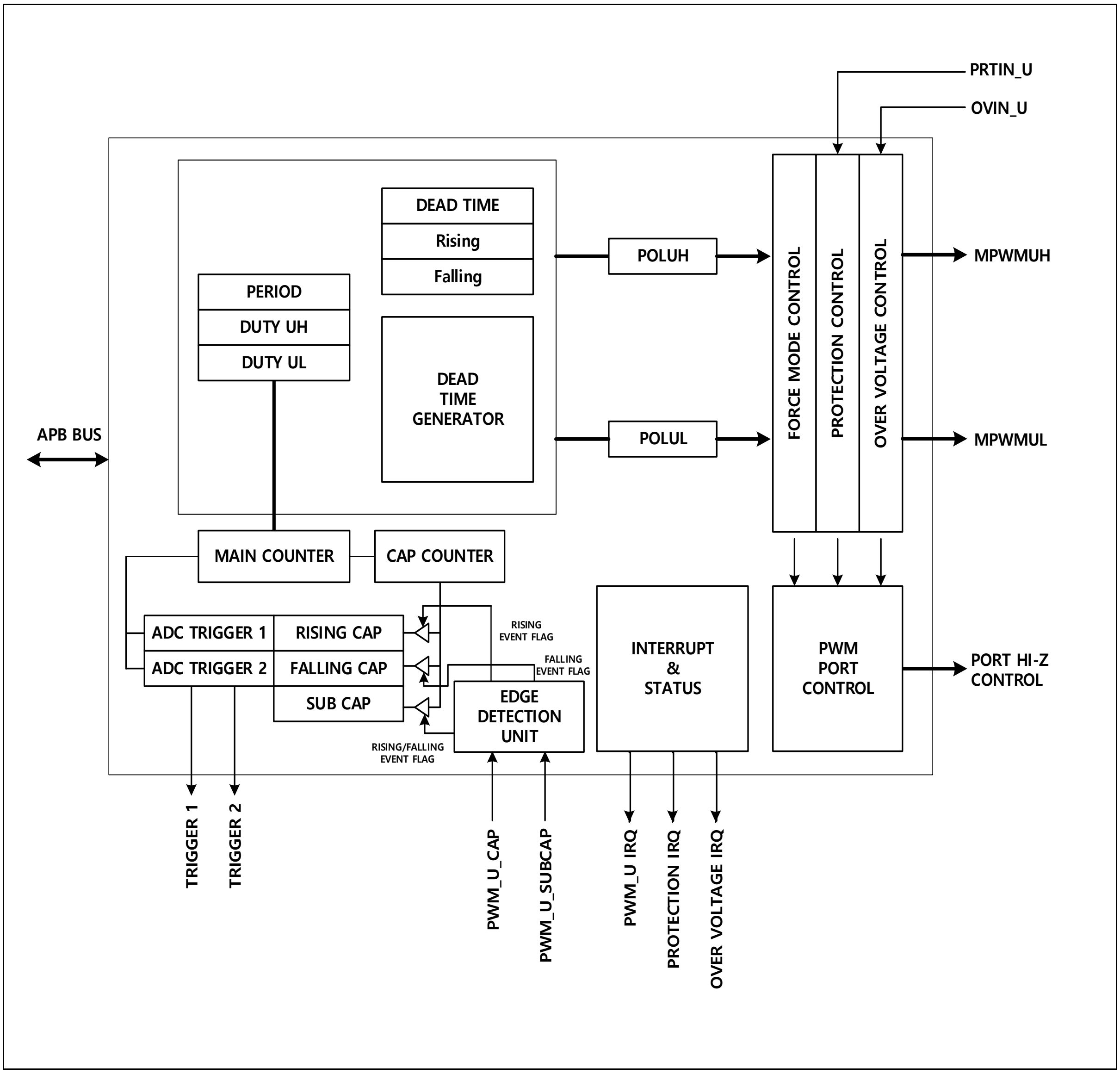


Figure 97 describes MPWM in block diagram (detailed).

Figure 97. MPWM Block Diagram: Phase U



NOTE: The figure above represents the phase U block only. The phase V and W blocks are identical to the phase U block. Each phase block is independently configurable; however, the protection and overvoltage interrupts are used in common.

14.2 Registers

Base address of MPWM is introduced in the followings:

Table 58. Base Address of MPWM Interface

Name	Base address
MPWM0	0x4000_4000
MPWM1	0x4000_5000

Table 59. MPWM Register Map

Name	Offset	Type	Description	Reset value	Reference
MPWMn_MR	0x0000	RW	MPWM n mode register	0x0000_0000	14.2.1
MPWMn_OLR	0x0004	RW	MPWM n output level register	0x0000_0000	14.2.2
MPWMn_FOLR	0x0008	RW	MPWM n forced output register	0x0000_0000	14.2.3
MPWMn_PRD	0x000C	RW	MPWM n PWM period register	0x0000_0002	14.2.4
MPWMn_DUH	0x0010	RW	MPWM n duty UH register	0x0000_0001	14.2.5
MPWMn_DVH	0x0014	RW	MPWM n duty VH register	0x0000_0001	14.2.6
MPWMn_DWH	0x0018	RW	MPWM n duty WH register	0x0000_0001	14.2.7
MPWMn_DUL	0x001C	RW	MPWM n duty UL register	0x0000_0001	14.2.8
MPWMn_DVL	0x0020	RW	MPWM n duty VL register	0x0000_0001	14.2.9
MPWMn_DWL	0x0024	RW	MPWM n duty WL register	0x0000_0001	14.2.10
MPWMn_CR1	0x0028	RW	MPWM n control register 1	0x0000_0000	14.2.11
MPWMn_CR2	0x002C	RW	MPWM n control register 2	0x0000_0000	14.2.12
MPWMn_SR	0x0030	RW	MPWM n status register	0x0000_0000	14.2.13
MPWMn_IER	0x0034	RW	MPWM n interrupt enable register	0x0000_0000	14.2.14
MPWMn_CNT	0x0038	R	MPWM n counter register	0x0000_0000	14.2.15
MPWMn_DTR	0x003C	RW	MPWM n dead time register	0x0000_0000	14.2.16
MPWMn_PCR	0x0040	RW	MPWM n protection register	0x0000_0000	14.2.17
MPWMn_PSR	0x0044	RW	MPWM n protection status register	0x0000_0000	14.2.18
MPWMn_OCR	0x0048	RW	MPWM n overvoltage detection register	0x0000_0000	14.2.19
MPWMn_OSR	0x004C	RW	MPWM n overvoltage detection status register	0x0000_0000	14.2.20
MPWMn_ATR1	0x0058	RW	MPWM n ADC trigger 1 register	0x0000_0000	14.2.21
MPWMn_ATR2	0x005C	RW	MPWM n ADC trigger 2 register	0x0000_0000	14.2.21
MPWMn_ATR3	0x0060	RW	MPWM n ADC trigger 3 register	0x0000_0000	14.2.21
MPWMn_ATR4	0x0064	RW	MPWM n ADC trigger 4 register	0x0000_0000	14.2.21
MPWMn_ATR5	0x0068	RW	MPWM n ADC trigger 5 register	0x0000_0000	14.2.21
MPWMn_ATR6	0x006C	RW	MPWM n ADC trigger 6 register	0x0000_0000	14.2.21

Table 59. MPWM Register Map (continued)

Name	Offset	Type	Description	Reset value	Reference
MPWMn_CPCR	0x0070	RW	MPWM n comparator protection control register	0x0000_0000	14.2.22
MPWMn_CR3	0x0080	RW	MPWM n control register 3	0x0000_0000	14.3.1
MPWMn_CR4	0x0084	RW	MPWM n control register 4	0x0000_0000	14.3.2
MPWMn_PRDU	0x0090	RW	MPWM n phase U period register	0x0000_0002	14.3.3
MPWMn_PRDV	0x0094	RW	MPWM n phase V period register	0x0000_0002	14.3.4
MPWMn_PRDW	0x0098	RW	MPWM n phase W period register	0x0000_0002	14.3.5
MPWMn_CNTU	0x00A0	RO	MPWM n phase U counter register	0x0000_0000	14.3.6
MPWMn_CNTV	0x00A4	RO	MPWM n phase V counter register	0x0000_0000	14.3.7
MPWMn_CNTW	0x00A8	RO	MPWM n phase W counter register	0x0000_0000	14.3.8
MPWMn_DTRU	0x00B0	RW	MPWM n phase U dead time register	0x0000_0000	14.3.9
MPWMn_DTRV	0x00B4	RW	MPWM n phase V dead time register	0x0000_0000	14.3.10
MPWMn_DTRW	0x00B8	RW	MPWM n phase W dead time register	0x0000_0000	14.3.11
MPWMn_CAPCNTU	0x00C0	RW	MPWM n phase U capture counter register	0x0000_0001	14.3.12
MPWMn_CAPCNTV	0x00C4	RW	MPWM n phase V capture counter register	0x0000_0001	14.3.12
MPWMn_CAPCNTW	0x00C8	RW	MPWM n phase W capture counter register	0x0000_0001	14.3.12
MPWMn_RCAPU	0x00D0	RW	MPWM n phase U rising capture register	0x0000_0000	14.3.13
MPWMn_RCAPV	0x00D4	RW	MPWM n phase V rising capture register	0x0000_0000	14.3.13
MPWMn_RCAPW	0x00D8	RW	MPWM n phase W rising capture register	0x0000_0000	14.3.13
MPWMn_FCAPU	0x00E0	RW	MPWM n phase U falling capture register	0x0000_0000	14.3.14
MPWMn_FCAPV	0x00E4	RW	MPWM n phase V falling capture register	0x0000_0000	14.3.14
MPWMn_FCAPW	0x00E8	RW	MPWM n phase W falling capture register	0x0000_0000	14.3.14
MPWMn_SCAPU	0x00F0	RW	MPWM n phase U sub-capture register	0x0000_0000	14.3.15
MPWMn_SCAPV	0x00F4	RW	MPWM n phase V sub-capture register	0x0000_0000	14.3.15
MPWMn_SCAPW	0x00F8	RW	MPWM n phase W sub-capture register	0x0000_0000	14.3.15

14.2.1 MPWMn_MR: MPWM n mode register

MPWMn_MR is a 16-bit register.

MPWM0_MR=0x4000_4000, MPWM1_MR=0x4000_5000

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	MOTORB	Reserved	UAO	Reserved	TUP	BUP	Reserved	MCHMOD	UPDOWN
-	00	-	0	-	0	0	-	00	0
-	RW	-	RW	-	RW	RW	-	RW	RW

15 14	MOTORB	MPWM mode selection	
		00	MPWM mode
		01	Normal PWM mode
		11	Individual PWM mode
7	UAO	Update timing setting	
		0	Updates are performed at designated timings.
		1	The duty and period are updated immediately when requested. With this setting, the duty and period registers are updated after two MPWM clock cycles.
5	TUP	Whether to enable or disable period/duty updates (at period matches)	
		0	Periods and duties are not updated at every period match.
		1	Periods and duties are updated at every period match.
4	BUP	Whether to enable or disable period/duty updates (at bottom matches)	
		0	Periods and duties are not updated at every bottom match.
		1	Periods and duties are updated at every bottom match.
2 1	MCHMOD	Channel symmetry/asymmetry mode selection (In Normal PWM mode, it operates without setting)	
		00	Two-channel symmetric mode The H-side duty determines the H channel signal's timing of switching between the high and low levels. The L-side duty determines the L channel signal's timing of switching between the high and low levels.
		01	One-channel asymmetric mode The H-side duty determines the H channel signal's timing of switching to the high level. The L-side duty determines the H channel signal's timing of switching to the low level. The H and L channels are inverted.
		10	One-channel symmetric mode The H-side duty determines the H channel signal's timing of switching between the high and low levels. The H and L channels are inverted.
		11	Reserved
0	UPDOWN	PWM up/down counter mode selection	
		0	PWM up-counter mode (used when MOTORB = 1)
		1	PWM up-down counter mode (used when MOTORB = 0, 1, or 3)

NOTE: In Individual PWM mode, MPWM0 and MPWM1 cannot be used simultaneously.

After initial PWM period and duty setting is completed, the UAO bit must be set once to update internal operating registers with these set values. This helps transfer the setup data from the user interface register to internal operating registers. The setting of the UAO bit must remain unchanged for at least two PWM clock cycles; otherwise, the update command may get lost, resulting in retaining old data in the internal registers.

The MCHMOD value of MR is valid only when the MOTORB value of MR is 0 or 3; otherwise, the MCHMOD field value is internally ignored and remains at “00.”

The UPDOWN value of the MR field is valid if the MOTORB of MR has been set to 1; otherwise, the UPDOWN value is internally ignored and remains at “1.” The PWM counter always operates as an up-down counter both in MPWM mode and in Individual PWM mode.

14.2.2 MPWMn_OLR: MPWM n output level register

MPWMn_OLR is a 32-bit register that controls the level of each PWM output port.

MPWM0_OLR=0x4000_4004, MPWM1_OLR=0x4000_5004

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																		DOLWH	DOLVH	DOLUH	DOLWL	DOLVL	DOLUL	Reserved		WHL	VHL	UHL	WLL	VLL	ULL
-																		0	0	0	0	0	0	-		0	0	0	0	0	0
-																		RW	RW	RW	RW	RW	RW	-		RW	RW	RW	RW	RW	RW

13	DOLWH	Selection of the output level when PWMxH is set disabled (x = U, V, W)	
		0	Low level
		1	High level
12	DOLVH	0	Low level
		1	High level
11	DOLUH	0	Low level
		1	High level
10	DOLWL	Selection of the output level when PWMxL is set disabled (x = U, V, W)	
		0	Low level
		1	High level
9	DOLVL	0	Low level
		1	High level
8	DOLUL	0	Low level
		1	High level
5	WHL	Whether or not to invert the output at the start of PWMxH (x = U, V, W)	
		0	Basic output level (low)
		1	Inverted output level (high)
4	VHL	0	Basic output level (low)
		1	Inverted output level (high)
3	UHL	0	Basic output level (low)
		1	Inverted output level (high)
2	WLL	Whether or not to invert the output at the start of PWMxL (x = U, V, W)	
		0	Basic output level (low)
		1	Inverted output level (high)
1	VLL	0	Basic output level (low)
		1	Inverted output level (high)
0	ULL	0	Basic output level (low)
		1	Inverted output level (high)

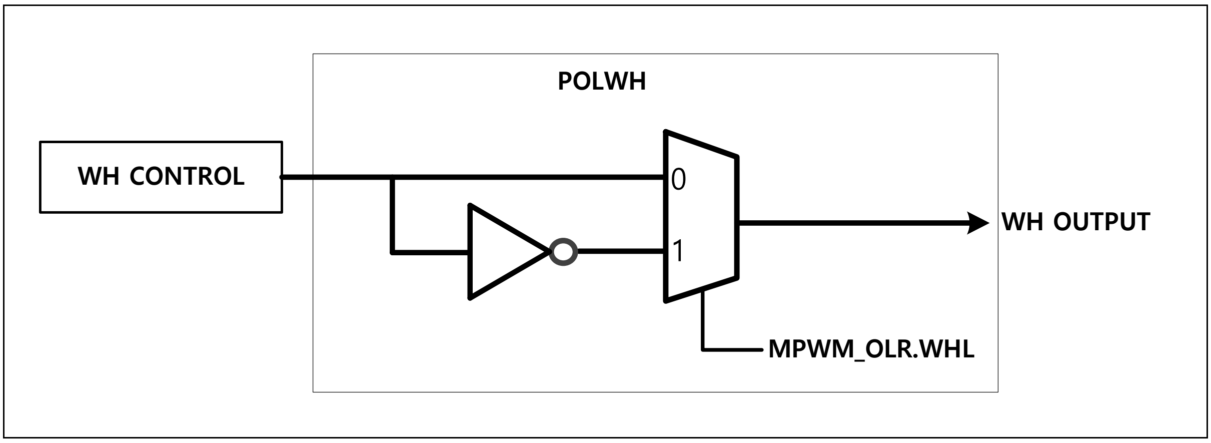
NOTE: Refer to Table 60 on the following page for the basic output levels in each operating mode. DOLxH and DOLxL(x=U,V,W) refers to the output level when the PWM output has been stopped.

Table 60. MPWM Basic Output Level: MPWMn_OLR = 0x00

PWM Output	Level	Normal PWM mode (MOTORB = 1)		MPWM mode (MOTORB = 0)
		Up mode (UPDOWN = 0)	Up-down mode (UPDOWN = 1)	
WH	Basic output level	LOW	HIGH	HIGH
	Inverted output level	HIGH	LOW	LOW
WL	Basic output level	LOW	LOW	LOW
	Inverted output level	HIGH	HIGH	HIGH
VH	Basic output level	LOW	HIGH	HIGH
	Inverted output level	HIGH	LOW	LOW
VL	Basic output level	LOW	LOW	LOW
	Inverted output level	HIGH	HIGH	HIGH
UH	Basic output level	LOW	HIGH	HIGH
	Inverted output level	HIGH	LOW	LOW
UL	Basic output level	LOW	LOW	LOW
	Inverted output level	HIGH	HIGH	HIGH

The figure below illustrates the polarity control block. This is an example for the WH signal polarity control block.

Figure 98. Polarity Control Block



14.2.3 MPWMn_FOLR: MPWM n forced output level register

MPWMn_FOLR is an 8-bit register. A specific PWM output level can be forcibly generated by an abnormality event triggered by an external condition or the user’s intentional manipulation. Once a forcing condition is met, each PWM channel generates an output signal in the level set in the FOLR register.

MPWM0_FOLR=0x4000_4008, MPWM1_FOLR=0x4000_5008

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																										WHFL	VHFL	UHFL	WLFL	VLFL	ULFL
-																										0	0	0	0	0	0
-																										RW	RW	RW	RW	RW	RW

5	WHFL	WH forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”
4	VHFL	VH forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”
3	UHFL	UH forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”
2	WLFL	WL forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”
1	VLFL	VL forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”
0	ULFL	UL forced output level selection	
		0	Output force level: “L”
		1	Output force level: “H”

14.2.4 MPWMn_PRD: MPWM n period register

MPWMn_PRD is a 32-bit register.

MPWM0_PRDU=0x4000_400C, MPWM1_PRDU=0x4000_500C

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		PERIOD	
-		0x0002	
-		RW	

15 0	PERIOD	16-bit PWM period of phases U, V, and W. The bit value must be larger than 0x0010.

14.2.5 MPWMn_DUH: MPWM n duty UH register

MPWMn_DUH is a 32-bit register.

MPWM0_DUH=0x4000_4010, MPWM1_DUH=0x4000_5010

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		DUTY UH	
-		0x0001	
-		RW	

15 0	DUTY UH	16-bit PWM duty for UH output (The duty can be set to 0.)
NOTE: Setting a full duty generates a “low” output, while setting a zero duty generates a “high” output		

14.2.6 MPWMn_DVH: MPWM n duty VH register

MPWMn_DVH is a 32-bit register.

MPWM0_DVH=0x4000_4014, MPWM1_DVH=0x4000_5014

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		DUTY VH	
-		0x0001	
-		RW	

15 0	DUTY VH	16-bit PWM duty for VH output (The duty can be set to 0.)

NOTE: Setting a full duty generates a “low” output, while setting a zero duty generates a “high” output

14.2.7 MPWMn_DWH: MPWM n duty WH register

MPWMn_DWH is a 32-bit register.

MPWM0_DWH=0x4000_4018, MPWM1_DWH=0x4000_5018

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		DUTY WH	
-		0x0001	
-		RW	

15 0	DUTY WH	16-bit PWM duty for WH output (The duty can be set to 0.)

NOTE: Setting a full duty generates a “low” output, while setting a zero duty generates a “high” output

14.2.8 MPWMn_DUL: MPWM n duty UL register

MPWMn_DUL is a 32-bit register.

MPWM0_DUL=0x4000_401C, MPWM1_DUL=0x4000_501C

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	DUTY UL
-	0x0001
-	RW

15 0	DUTY UL	16-bit PWM duty for UL output (The duty can be set to 0.)

NOTE: Setting a full duty generates a "low" output, while setting a zero duty generates a "high" output

14.2.9 MPWMn_DVL: MPWM n duty VL register

MPWMn_DVL is a 32-bit register.

MPWM0_DVL=0x4000_4020, MPWM1_DVL=0x4000_5020

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	DUTY VL
-	0x0001
-	RW

15 0	DUTY VL	16-bit PWM duty for VL output (The duty can be set to 0.)

NOTE: Setting a full duty generates a "low" output, while setting a zero duty generates a "high" output

14.2.10 MPWMn_DWL: MPWM n duty WL register

MPWMn_DWL is a 32-bit register.

MPWM0_DWL=0x4000_4024, MPWM1_DWL=0x4000_5024

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		DUTY WL	
-		0x0001	
-		RW	

15 0	DUTY WL	16-bit PWM duty for WL output (The duty can be set to 0.)
NOTE: Setting a full duty generates a “low” output, while setting a zero duty generates a “high” output		

14.2.11 MPWMn_CR1: MPWM n control register 1

MPWMn_CR1 is a 16-bit register.

MPWM0_CR1=0x4000_4028, MPWM1_CR1=0x4000_5028

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			IRQN	Reserved	PWMEN
-			000	-	0
-			RW	-	RW

10 8	IRQN	Interrupt request (IRQ) interval number (PRDIRQ, BOTIRQ, and ATRn are made at the specified intervals)
0	PWMEN	Whether to enable or disable the PWM (0: disables, 1: enables)
		If this bit is set to 0, the PWM block remains in the reset state but the user interface is accessible. This bit must be set to 1 to operate the PWM block.
NOTE:	By default, PRDIRQ and BOTIRQ are made at every period. However, interrupt intervals can be set from 0 to 8 periods. If IRQN.CR1 = 0, IRQ is made at every period. Otherwise, they are made at every “IRQN + 1” periods.	

14.2.12 MPWMn_CR2: MPWM n control register 2

MPWMn_CR2 is a 16-bit register.

MPWM0_CR2=0x4000_402C, MPWM1_CR2=0x4000_502C

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																								HALT	Reserved						PSTART
-																								0	-						0
-																								RW	-						RW

7	HALT	Setting of PWM halting, which stops the PWM counter but does not reset it. If HALT = 1, the PWM output retains its state when a halt is made.	
0	PSTART	0	Stops and clears the PWM counter.
		1	Starts the PWM counter (re-initialized at the second PWM clock period).

NOTE: Before setting PSTART, PWMEN must be set to 1 to start the PWM counter.

14.2.13 MPWMn_SR: MPWM n status register

MPWMn_SR is a 32-bit register.

MPWM0_SR=0x4000_4030, MPWM1_SR=0x4000_5030

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	WDOWN	WIRQCNT	VDOWN	VIRQCNT	UDOWN/DO	UIRQCNT /IRQCNT	PRDWIF	BOTWIF	PRDVIF	BOTVIF	PRDUIF	BOTUIF	DWHIF/ATR6	DVHIF/ATR5	DUHIF/ATR4	DWLIF/ATR3	DVLIF/ATR2F	DULIF/ATR1F
-	0	000	0	000	0	000	0	0	0	0	0	0	0	0	0	0	0	0
-	RO	RO	RO	RO	RO	RO	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC

23	WDOWN	Current mode of PWM's phase W counter 0: current counter mode = up-counter 1: current counter mode = down-counter	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays which mode the phase W counter is in.
22 20	WIRQCNT	Phase W channel's period match interrupt count value (Interval PRDIRQ mode)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the phase W channel's IRQ interrupt count value.
19	VDOWN	Current mode of PWM's phase V counter 0: current counter mode = up-counter 1: current counter mode = down-counter	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays which mode the phase V counter is in.
18 16	VIRQCNT	Phase V channel's period match interrupt count value (Interval PRDIRQ mode)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the phase V channel's IRQ interrupt count value.
15	DOWN UDOWN	Current PWM counter mode 0: current counter mode = up-counter 1: current counter mode = down-counter	
		Motor mode	The bit displays which mode the PWM counter is in.
		Individual mode	The bit displays which mode the phase U counter is in.
14 12	IRQCNT UIRQCNT	All channels' or phase U channel's period match interrupt count value (Interval PRDIRQ mode)	
		Motor mode	The bit displays the phase U, V, and W channels' interrupt count value.
		Individual mode	The bit displays the phase U channel's IRQ interrupt count value.
11	PRDWIF	PWM period interrupt flag (Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase W channel's period interrupt flag
10	BOTWIF	PWM bottom interrupt flag (Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase W channel's bottom interrupt flag
9	PRDVIF	PWM period interrupt flag	

		(Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase V channel’s period interrupt flag
8	BOTVIF	PWM bottom interrupt flag (Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase V channel’s bottom interrupt flag
7	PRDUIF	PWM period interrupt flag (Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	Period interrupt flag
		Individual mode	Phase U channel’s period interrupt flag
6	BOTUIF	PWM bottom interrupt flag (Writing a 1 to the bit clears the flag) (0: The interrupt has not occurred, 1: The interrupt has occurred)	
		Motor mode	Bottom interrupt flag
		Individual mode	Phase U channel’s bottom interrupt flag
5	DWHIF ATR6F	PWM duty WH interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR6 is set disabled.)	
		0	The interrupt has not occurred.
		1	The interrupt has occurred.
4	DVHIF ATR5F	PWM duty VH interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR5 is set disabled.)	
		0 1	The interrupt has not occurred. The interrupt has occurred.
3	DUHIF ATR4F	PWM duty UH interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR4 is set disabled.)	
		0 1	The interrupt has not occurred. The interrupt has occurred.
2	DWLIF ATR3F	PWM duty WL interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR3 is set disabled.)	
		0 1	The interrupt has not occurred. The interrupt has occurred.
1	DVLIF ATR2F	PWM duty VL interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR2 is set disabled.)	
		0 1	The interrupt has not occurred. The interrupt has occurred.
0	DULIF ATR1F	PWM duty UL interrupt flag (Writing a 1 to the bit clears the flag) (The duty interrupt is enabled when ATR1 is set disabled.)	
		0 1	The interrupt has not occurred. The interrupt has occurred.

NOTE: Each of the MPWM_SR [5: 0] status bits is shared by a duty match interrupt event or an ADC trigger match interrupt event. If the ADC trigger mode is set disabled, these bits are flagged by duty match interrupts; otherwise, they are flagged by ADC trigger counter match interrupts. The ADC trigger mode is selected in the ATMOD bit field of the ATRm register.

14.2.14 MPWMn_IER: MPWM n interrupt enable register

MPWMn_ IER is a 32-bit register.

MPWM0_IER=0x4000_4034, MPWM1_IER=0x4000_5034

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																				PRDWIE	BOTWIE	PRDVIE	BOTVIE	PRDIE/PRDUIE	BOTIE/BOTUIE	WHIE	VHIE	UHIE	WLIE	VLIE	ULIE
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

11	PRDWIE	Whether to enable or disable the PWM's phase W counter period interrupt (0: disables, 1: enables)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase W channel's period interrupt
10	BOTWIE	Whether to enable or disable the PWM's phase W counter bottom interrupt (0: disables, 1: enables)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase W channel's bottom interrupt
9	PRDVIE	Whether to enable or disable the PWM's phase V counter period interrupt (0: disables, 1: enables)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase V channel's period interrupt
8	BOTVIE	Whether to enable or disable the PWM's phase V counter bottom interrupt (0: disables, 1: enables)	
		Motor mode	The bit is not used in this mode.
		Individual mode	Phase V channel's bottom interrupt
7	PRDIE PRDUIE	Whether to enable or disable the PWM counter's or the phase U counter's period interrupt (0: disables, 1: enables)	
		Motor mode	Phase U, V, and W channels' period interrupt
		Individual mode	Phase U channel's period interrupt
6	BOTIE BOTUIE	Whether to enable or disable the PWM counter's or the phase U counter's bottom interrupt (0: disables, 1: enables)	
		Motor mode	Phase U, V, and W channels' bottom interrupt
		Individual mode	Phase U channel's bottom interrupt
5	WHIE ATR6IE	Whether to enable or disable the WH duty or ATR6 match interrupt	
		0	Disables the interrupt.
		1	Enables the interrupt.
4	VHIE ATR5IE	Whether to enable or disable the VH duty or ATR5 match interrupt	
		0	Disables the interrupt.
		1	Enables the interrupt.
3	UHIE ATR4IE	Whether to enable or disable the UH duty or ATR4 match interrupt	
		0	Disables the interrupt.
		1	Enables the interrupt.
2	WLIE ATR3IE	Whether to enable or disable the WL duty or ATR3 match interrupt	
		0	Disables the interrupt.
		1	Enables the interrupt.
1	VLIE	Whether to enable or disable the VL duty or ATR2 match interrupt enable bit	

	ATR2IE	0	Disables the interrupt.
		1	Enables the interrupt.
0	ULIE ATR1IE	Whether to enable or disable the UL duty or ATR1 match interrupt enable bit	
		0	Disables the interrupt.
		1	Enables the interrupt.

NOTES:

1. Each of the MPWM.IER [5: 0] status bits is shared by a duty match interrupt event or an ADC trigger match interrupt event.
2. If the ADC trigger mode is set disabled, these bits are flagged by duty match interrupts; otherwise, they are flagged by ADC trigger counter match interrupts.
3. The ADC trigger mode is selected in the ATMOD bit field of the ATRm register. In Individual mode, phases U, V, and W have different interrupt vectors.
4. For example, if ULI, UHI, BOTU, or PRDU is flagged, the MPWM_U interrupt occurs.
5. Additionally, ATR1 and ATR2 are used as MPWM_U interrupts; ATR3 and ATR4 are used as MPWM_V interrupts; and ATR5 and ATR6 are used as MPWM_W interrupts.

14.2.15 MPWMn_CNT: MPWM n counter register

MPWMn_CNT is a 32-bit read-only register.

MPWM0_CNTU=0x4000_4038, MPWM1_CNTU=0x4000_5038

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CNT</td></tr>
<tr><td colspan="16">-</td><td colspan="16">0x0000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RO</td></tr>
</table>

15 0	CNT	PWM counter value The main counter value of the PWM

14.2.16 MPWMn_DTR: MPWM n dead time register 2

MPWMn_DTR is a 32-bit register. Its settings are applied to phases U, V, and W simultaneously in motor mode, and only to phase U in Individual mode.

MPWM0_DTRU=0x4000_403C, MPWM1_DTRU=0x4000_503C

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	DTEN	PSHRT	DTMDSEL	Reserved	DTCLK	DT
-	0	0	0	-	0	0x00
-	RW	RW	RW	-	RW	RW

15	DTEN	Whether to enable or disable the dead time function You must disable this function in two-channel symmetric mode because this mode does not support the function. 0: Disables the dead time function. 1: Enables the dead time function.
14	PSHRT	Short-circuit protection This function is supported in two-channel symmetric mode. In one-channel mode, both the H and L sides are not simultaneously active. When either side is active, the other side always inactive. 0: Enables short-circuit protection (Automatically ensures that both the H and L sides are not simultaneously active) 1: Disables short-circuit protection
13	DTMDSEL	Dead time mode selection When the POL function is used, a phase's H and L signals can overlap in the high level. This bit is typically set to prevent such a phenomenon. 0: Inserts dead time at the leading edge of PWMxH and the trailing edge of PWMxL 1: Inserts dead time at the trailing edge of PWMxH and the leading edge of PWMxL
9 8	DTCLK	Dead time prescaler 00: Sets PWM CLK/2 as the dead time counter clock. 01: Sets PWM CLK/4 as the dead time counter clock. 10: Sets PWM CLK/8 as the dead time counter clock. 11: Sets PWM CLK/16 as the dead time counter clock.
7 0	DT	Rising or falling edge dead time value (set as the delay time from the normal polarity to the falling edge level output) 0x01 through 0xFF: dead time.

NOTE: Writing a 0 to the bit will have the same effect as disabling the dead time.

NOTE: Values written to this register are applied to the rising or falling edge dead times of all three phases U, V, and W. To set a particular phase's dead time, you must configure Individual mode's registers DTRU, DTRV, and DTRW. This short-circuit protection is applied to the PWM's internal signals and not to its external signals. When the internal signals from the H and L sides are all in the high level, short-circuit protection induces both sides to generate low-level outputs.

14.2.17 MPWMn_PCR: MPWM n protection control register

MPWMn_PCR is a 32-bit register.

MPWM0_PCR=0x4000_4040, MPWM1_PCR=0x4000_5040

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
WPROTEN	WPROTPOL	Reserved			WPROTD			VPROTEN	VPROTPOL	Reserved			VPROTD			PROTEN	PROTPOL	Reserved			PROTD			PROTIE	Reserved	WHPROTM	VHPROTM	UHPROTM	WLPROTM	VLPROTM	ULPROTM
0	0	-			000			0	0	-			000			0	0	-			000			0	-	0	0	0	0	0	0
RW	RW	-			RW			RW	RW	-			RW			RW	RW	-			RW			RW	-	RW	RW	RW	RW	RW	RW

31	WPROTEN	Whether to enable or disable protection input 0: Disables. 1: Enables.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W protection input.
30	WPROTPOL	Protection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase W protection input.
26 24	WPROTD	Protection input debounce 0: No debounce 1–7: Debounce time is set to 1/MPWMCLK * PROTD[2:0]	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W protection input pin debounce.
23	VPROTEN	Whether to enable or disable protection input 0: Disables. 1: Enables.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase V protection input.
22	VPROTPOL	Protection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase V protection input.
18 16	VPROTD	Protection input debouncing 0: No debouncing 1–7: Debounce time is set to 1/MPWMCLK * PROTD[2:0]	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase V protection input pin debounce.
15	PROTEN UPROTEN	Whether to enable or disable protection input 0: Disables. 1: Enables.	
		Motor mode	The bit enables PWM protection input.
		Individual mode	The bit enables phase U protection input.
14	PROTPOL	Protection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit selects the polarity of PWM protection input.
		Individual mode	The bit selects the polarity of phase U protection input.

10 8	PROTD	Protection input debouncing 0: No debouncing 1–7: Debounce time is set to 1/MPWMCLK * PROTD[2:0]	
		Motor mode	The bit enables PWM protection input pin debounce.
		Individual mode	The bit enables phase U protection input pin debounce.
7	PROTIE	Whether to enable or disable the protection interrupt	
		0 1	Disables the protection interrupt. Enables the protection interrupt.
5	WHPROTM	Whether to enable or disable protection output at the H side of phase W.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
4	VHPROTM	Whether to enable or disable protection output at the H side of phase V.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
3	UHPROTM	Whether to enable or disable protection output at the H side of phase U.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
2	WLPROTM	Whether to enable or disable protection output at the L side of phase W.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
1	VLPROTM	Whether to enable or disable protection output at the L side of phase V.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
0	ULPROTM	Whether to enable or disable protection output at the L side of phase U.	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
NOTE:	To enable protection output at the H side of phase U in Individual mode, you must first enable phase U protection input. And there is only one protection interrupt vector in Individual mode as well. You can control the vector by checking the interrupt flag in the interrupt register.		

14.2.18 MPWMn_PSR: MPWM n protection status register

MPWMn_PSR is a 16-bit register that displays the protection status. Any value written to this register without writing to PROTKEY is dismissed.

MPWM0_PSR=0x4000_4044, MPWM1_PSR=0x4000_5044

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		PROTKEY	PROTIF	Reserved	WHPROTF	VHPROTF	UHPROTF	WLPROTF	VLPROTF	ULPROTF
-		-	0	-	0	0	0	0	0	0
-		WO	RC	-	RW	RW	RW	RW	RW	RW

15 8	PROTKEY	Protection clear access key (OSR key: 0xCA) To clear a protection flag, you must write to the flag and this access key. Writing to a flag bit without writing to PROTKEY is prohibited.	
7	PROTIF	Protection interrupt status	
		0	The protection interrupt has not occurred.
		1	The protection interrupt has occurred (Write: Clears the flag).
5	WHPROT	Phase W’s H-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
4	VHPROT	Phase V’s H-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
3	UHPROT	Phase U’s H-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
2	WLPROT	Phase W’s L-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
1	VLPROT	Phase V’s L-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
0	ULPROT	Phase U’s L-side protection flag	
		0	Protection not occurred. (Write: Clears the flag)
		1	Protection has occurred (Write: Forcibly generates an output signal at the level set in FOLR)

NOTE: If an arbitrary signal is applied to an external protection pin when the PCR register’s corresponding PROTEN bit is enabled, the PWM output will be generated at the level set in FOLR. Additionally, the user can write to the PSR register to manually inhibit or forcibly generate a specific output.

14.2.19 MPWMn_OCR: MPWM n overvoltage control register

MPWMn_OCR is a 32-bit register.

MPWM0_OCR=0x4000_4048, MPWM1_OCR=0x4000_5048

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
WOVINEN	WOVINPOL	Reserved	WOVIND	VOVINEN	VOVINPOL	Reserved	VOVIND	OVINEN	OVINPOL	Reserved	OVIND	OVINIE	Reserved	WHOVINM	VHOVINM	UHOVINM	WLOVINM	VLOVINM	ULOVINM
0	0	-	000	0	0	-	000	0	0	-	000	0	-	0	0	0	0	0	0
RW	RW	-	RW	RW	RW	-	RW	RW	RW	-	RW	RW	-	RW	RW	RW	RW	RW	RW

31	WOVINEN	Whether to enable or disable overvoltage detection input 0: Disables. 1: Enables.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W overvoltage detection input.
30	WOVINPOL	Overvoltage detection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase W overvoltage detection input.
26 24	WOVIND	Overvoltage detection input debounce 0: No debounce 1–7: Debounce time is set to 1/MPWMCLK * OVIND[2:0]	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W overvoltage detection input debounce.
23	VOVINEN	Whether to enable or disable overvoltage detection input 0: Disables. 1: Enables.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W overvoltage detection input.
22	VOVINPOL	Overvoltage detection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase V overvoltage detection input.
18 16	VOVIND	Overvoltage detection input debounce 0: No debounce 1–7: Debounce time is set to 1/MPWMCLK * OVIND[2:0]	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase V overvoltage detection input debounce.
15	OVINEN UOVINEN	Whether to enable or disable overvoltage detection input 0: Disables.	

		1: Enables.	
		Motor mode	The bit enables PWM overvoltage detection input.
		Individual mode	The bit enables phase U overvoltage detection input.
14	OVINPOL	Overvoltage detection input polarity selection 0: L-active 1: H-active	
		Motor mode	The bit selects the polarity of PWM overvoltage detection input.
		Individual mode	The bit selects the polarity of phase U overvoltage detection input.
10 8	OVIND	Overvoltage detection input debounce 0: No debounce 1–7: Debounce time is set to 1/MPWMCLK * OVIND[2:0]	
		Motor mode	The bit enables PMW overvoltage detection input debounce.
		Individual mode	The bit enables phase U overvoltage detection input debounce.
7	OVINIE	Whether to enable or disable the overvoltage detection interrupt	
		0 1	Disables the protection interrupt. Enables the protection interrupt.
5	WHOVINM	Whether to enable or disable overvoltage detection output at the H side of phase W	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
4	VHOVINM	Whether to enable or disable overvoltage detection output at the H side of phase V	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
3	UHOVINM	Whether to enable or disable overvoltage detection output at the H side of phase U	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
2	WLOVINM	Whether to enable or disable overvoltage detection output at the L side of phase W	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
1	VLOVINM	Whether to enable or disable overvoltage detection output at the L side of phase V	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.
0	ULOVINM	Whether to enable or disable overvoltage detection output at the L side of phase U	
		0 1	Disables protection output. Enables the protection output at the level set in FOLR.

NOTE: To enable protection output at the H side of phase U in Individual mode, you must first enable phase U protection input. And there is only one protection interrupt vector in Individual mode as well. You can control the vector by checking the interrupt flag in the interrupt register.

14.2.20 MPWMn_OSR: MPWM n overvoltage status register

MPWMn_OSR is a 16-bit register that displays the overvoltage detection status. Any value written to this register without writing to OVINKEY is dismissed.

MPWM0_OSR=0x4000_404C, MPWM1_OSR=0x4000_504C

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		OVINKEY	OVINIF	Reserved	WHOVINF	VHOVINF	UHOVINF	WLOVINF	VLOVINF	ULOVINF
-		-	0	-	0	0	0	0	0	0
-		WO	RC	-	RW	RW	RW	RW	RW	RW

15 8	OVINKEY	Overvoltage detection clear access key (OSR key: 0xAC) To clear a protection flag, you must write to the flag and this access key. Writing to a flag bit without writing to OVINKEY is prohibited.	
7	OVINIF	Overvoltage detection interrupt status	
		0	The overvoltage interrupt has not occurred.
		1	The overvoltage interrupt has occurred (Write: Clears the flag).
5	WHOVIN	Phase W’s H-side overvoltage detection flag	
		0	Overvoltage has not occurred. (Write: Clears the flag)
		1	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
4	VHOVIN	Phase V’s H-side overvoltage detection flag	
		1	Overvoltage has not occurred. (Write: Clears the flag)
		0	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
3	UHOVIN	Phase U’s H-side overvoltage detection flag	
		0	Overvoltage has not occurred. (Write: Clears the flag)
		1	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
2	WLOVIN	Phase W’s L-side overvoltage detection flag	
		1	Over Voltage not occurred. (Write: Clears the flag)
		0	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
1	VLOVIN	Phase V’s L-side overvoltage detection flag	
		0	Over Voltage not occurred. (Write: Clears the flag)
		1	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)
0	ULOVIN	Phase U’s L-side overvoltage detection flag	
		1	Over Voltage not occurred. (Write: Clears the flag)
		0	Overvoltage has occurred (Write: Forcibly generates an output signal at the level set in FOLR)

NOTE: If an arbitrary signal is applied to an external protection pin when the OCR register’s corresponding OVINEN bit is enabled, the PWM output will be generated at the level set in FOLR. Additionally, the user can write to the OSR register to manually inhibit or forcibly generate a specific output.

14.2.21 MPWMn_ATRm: MPWM n ADC trigger counter register

MPWMn_ATRm is a 32-bit register.

MPWM0_ATR1=0x4000_4058, MPWM0_ATR2=0x4000_405C, MPWM0_ATR3=0x4000_4060
MPWM0_ATR4=0x4000_4064, MPWM0_ATR5=0x4000_4068, MPWM0_ATR6=0x4000_406C
MPWM1_ATR1=0x4000_5058, MPWM1_ATR2=0x4000_505C, MPWM1_ATR3=0x4000_5060
MPWM1_ATR4=0x4000_5064, MPWM1_ATR5=0x4000_5068, MPWM1_ATR6=0x4000_506C

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	ATSRC	Reserved	ATUDT	Reserved	ATMOD	ATCNT
-	00	-	0	-	0	0x0000
-	RW	-	RW	-	RW	RW

23 22	ATSRC	ADC trigger source counter	
		00	Sets phase U counter as the compare source (by default).
		01	Sets phase V counter as the compare source.
		10	Sets phase W counter as the compare source.
		11	Disables.
19	ATUDT	Trigger register update mode	
		0	ADC trigger value applied at period match event (at the same time with period and duty registers update)
		1	Trigger register update mode When this bit is set enabled, written trigger register values are sent to the trigger compare block after two PWM clock cycles (through synchronization logic)
17 16	ATMOD	ADC trigger mode register	
		00	ADC trigger Disable
		01	Trigger out when up count match
		10	Trigger out when down count match
		11	Trigger out when up-down count match
15 0	ATCNT	ADC trigger counter (The ADC trigger counter value must be smaller than the PWM period.) Setting the counter value to 0 prevents interrupts from being triggered.	

NOTE: In Individual mode, ATR1 and ATR2 are used as MPWM_U interrupts; ATR3 and ATR4 are used as MPWM_V interrupts; and ATR5 and ATR6 are used as MPWM_W interrupts.

14.2.22 MPWMn_CPCR: MPWM n comparator protection control register

MPWMn_CPCR is a 32-bit register.

MPWM0_CPCR=0x4000_4070, MPWM1_CPCR=0x4000_5070

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	WPCOMSEL	WCOMPROT	Reserved	VPCOMSEL	VCOMPROT	Reserved	PCOMSEL	COMPROT	Reserved
-	00	0	-	00	0	-	00	0	-
-	RW	RW	-	RW	RW	-	RW	RW	-

29 28	WPCOMSEL	Protection comparator peripheral selection 00: COMP0. 01: COMP1 10: COMP2 11: COMP3.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase W protection comparator peripheral.
27	WCOMPROT	Whether to enable or disable comparator protection input 0: Disables 1: Enables	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W comparator protection input.
21 20	VPCOMSEL	Protection comparator peripheral selection 00: COMP0. 01: COMP1 10: COMP2 11: COMP3.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase V protection comparator peripheral.
19	VCOMPROT	Whether to enable or disable comparator protection input 0: Disables 1: Enables	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase V comparator protection input.
13 12	PCOMSEL	Protection comparator peripheral selection 00: COMP0. 01: COMP1 10: COMP2 11: COMP3.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the polarity of phase U protection comparator peripheral.
11	COMPROT	Protection comparator peripheral selection 00: COMP0. 01: COMP1 10: COMP2	

	11: COMP3.	
	Motor mode	The bit is not used in this mode.
	Individual mode	The bit selects the polarity of phase U protection comparator peripheral.

NOTES:

1. To enable the Comparator protection output: MPWMn_CPCR.xCOMPROT (x: U, V, W) Bit and the MPWMn_PCR.xPROTEN (x: U, V, W) bit should all be set to '1'.
2. Protection In order to control the interrupt and polarity and each phase (U, V, W) protection output, it must be set in the MPWMn_PCR register.

14.3 Individual mode registers

14.3.1 MPWMn_CR3: MPWM n control register 3

To configure MPWMn_CR3, you must set the MOTORB bits in the MR register to 0x3 (Individual mode). This register replaces the CR1 and CR2 registers to set each phase individually for use in Individual mode.

If you one phase is used in individual mode, U-phase should be used. And when using 2 phases, you should use U and V phases.

MPWM0_CR3=0x4000_4080, MPWM1_CR3=0x4000_5080

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	WIRQN	WHALT	Reserved	WSTART	WEN	Reserved	VIRQN	VHALT	Reserved	VSTART	VEN	Reserved	UIRQN	UHALT	Reserved	USTART	UEN
-	000	0	-	0	0	-	000	0	-	0	0	-	000	0	-	0	0
-	RW	RW		RW	RW	-	RW	RW		RW	RW	-	RW	RW	-	RW	RW

22 20	WIRQN	Phase W interrupt interval setting (PRDIRQ, BOTIRQ, and ATRn are made at the specified intervals)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the IRQ intervals only for phase W.
NOTE: This bit field can be written to only when the WEN and UEN bit has been set to 1.			
19	WHALT	Whether or not to halt the phase W counter in the current state (Stops the counter clock only) Write: 0 = No effect, 1 = Halts the phase W counter Read: Phase W halt status (1 = halt)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit halts the phase W counter (together with the capture counter).
17	WSTART	PWM phase W start Writing to this bit has the counter recount from the beginning. Write: 0 = No effect, 1 = Starts the phase W counter Read: Phase W counter's start status (1 = start)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit starts the phase W count.
NOTE: This bit field can be written to only when the WEN bit has been set to 1.			
16	WEN	Whether to enable or disable PWM phase W Write: 0 = No effect, 1 = Enables phase W Read: Phase W enable status (1 = enabled)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase W.
14 12	VIRQN	Phase V interrupt interval setting (PRDIRQ, BOTIRQ, and ATRn are made at the specified intervals)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the IRQ intervals only for phase V.
NOTE: This bit field can be written to only when the VEN and UEN bit has been set to 1.			
11	VHALT	Whether or not to halt the phase V counter in the current state (Stops the counter clock only)	

		Write: 0 = No effect, 1 = Halts the phase V counter Read: Phase V halt status (1 = halt)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit halts the phase V counter (together with the capture counter).
9	VSTART	PWM phase V start Writing to this bit has the counter recount from the beginning. Write: 0 = No effect, 1 = Starts the phase V counter Read: Phase V start status (1 = start)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit starts the phase V count.
NOTE: This bit field can be written to only when the VEN bit has been set to 1.			
8	VEN	Whether to enable or disable PWM phase V Write: 0 = No effect, 1 = Enables phase V Read: Phase V enable status (1 = enabled)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables phase V.
6 4	UIRQN	Phase U interrupt interval setting (PRDIRQ, BOTIRQ, and ATRn are made at the specified intervals)	
		Motor mode	The bit sets the IRQ intervals for phases U, V, and W.
		Individual mode	The bit sets the IRQ intervals only for phase U.
NOTE: This bit field can be written to only when the UEN bit has been set to 1.			
3	UHALT	Whether or not to halt the phase U counter in the current state (Stops the counter clock only) Write: 0 = No effect, 1 = Halts the phase U counter Read: Phase U halt status (1 = halt)	
		Motor mode	The bit halts the phase U, V, and W counters.
		Individual mode	The bit halts the phase U counter (together with the capture counter).
1	USTART	PWM phase U start Writing to this bit has the counter recount from the beginning. Write: 0 = No effect, 1 = Starts the phase U counter Read: Phase U start status (1 = start)	
		Motor mode	The bit starts the phase U, V, and W counters.
		Individual mode	The bit starts the phase U counter.
0	UEN	Whether to enable or disable PWM phase U Write: 0 = No effect, 1 = Enables Read: Phase U enable status (1 = enabled)	
		Motor mode	The bit enables phases U, V, and W.
		Individual mode	The bit enables phase U.
NOTE: This bit field can be written to only when the UEN bit has been set to 1. By default, PRDIRQ and BOTIRQ are made at every period. However, interrupt intervals can be set from 1 to 8 periods. If IRQN.CR1 = 0, IRQ is made at every period. Otherwise, they are made at every “IRQN + 1” periods.			

14.3.2 MPWMn_CR4: MPWM n control register 4

To configure MPWMn_CR4, you must set the MOTORB bits in the MR register to 0x3 (Individual mode). This register replaces the CR1 and CR2 registers to set each phase individually for use in Individual mode.

MPWM0_CR4=0x4000_4084, MPWM1_CR4=0x4000_5084

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	WCONTI	Reserved	WSTOP	WDIS	Reserved	VCONTI	Reserved	VSTOP	VDIS	Reserved	UCONTI	Reserved	USTOP	UDIS
-	0	-	0	0	-	0	-	0	0	-	0	-	0	0
-	RW	-	RW	RW	-	RW		RW	RW	-	RW	-	RW	RW

19	WCONTI	Whether or not to resume the phase W counter Write: 0 = No effect, 1 = Resumes the phase W counter Read: Phase W halt status (1 = halt)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit resumes the phase U counter.
17	WSTOP	Whether or not to stop and reset the phase W counter Write: 0 = No effect, 1 = Stops and resets the phase W counter Read: Phase W counter start status (1 = start)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit stops and resets the phase W counter (together with the capture counter).
16	WDIS	Whether or not to disable and reset the phase W counter Write: 0 = No effect, 1 = Disables Read: Phase W enable status (1 = enabled)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit disables and resets the phase W counter (together with the capture counter).
11	VCONTI	Whether or not to resume the phase V counter Write: 0 = No effect, 1 = Resumes the phase V counter Read: Phase V halt status (1 = halt)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit resumes the phase V counter.
9	VSTOP	Whether or not to stop and reset the phase V counter Write: 0 = No effect, 1 = Stops and resets the phase V counter Read: Phase V counter start status (1 = start)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit stops and resets the phase V counter (together with the capture counter).
8	VDIS	Whether or not to disable and reset the phase V counter Write: 0 = No effect, 1 = Disables Read: Phase V enable status (1 = enabled)	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit disables and resets the phase V counter (together with the capture counter).

3	UCONTI	Whether or not to resume the phase U counter Write: 0 = No effect, 1 = Resumes the phase U counter Read: Phase U halt status (1 = halt)	
		Motor mode	The bit resumes the phase U, V, and W counters.
		Individual mode	The bit resumes the phase U counter.
1	USTOP	Whether or not to stop and reset the phase U counter Write: 0 = No effect, 1 = Stops and resets the phase U counter Read: Phase U counter start status (1 = start)	
		Motor mode	The bit stops and resets the phase U, V, and W counters.
		Individual mode	The bit stops and resets the phase U counter (together with the capture counter).
0	UDIS	Whether or not to disable and reset the phase U counter Write: 0 = No effect, 1 = Disables Read: Phase U enable status (1 = enabled)	
		Motor mode	The bit disables and resets the phase U, V, and W counters.
		Individual mode	The bit disables the phase U counter (together with the capture counter).

NOTE: Before setting PSTART, PWMEN must be set to 1 to start the PWM counter.

14.3.3 MPWMn_PRDU: MPWM n phase U period register

MPWMn_PRDU is a 32-bit register.

MPWM0_PRDU=0x4000_4090, MPWM1_PRDU=0x4000_5090

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		PERIOD_U	
-		0x0002	
-		RW	

15 0	PERIOD_U	16-bit PWM period The bit value must be larger than 0x0010.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the period for phase U only.

14.3.4 MPWMn_PRDV: MPWM n phase V period register

MPWMn_PRDV is a 32-bit register.

MPWM0_PRDV=0x4000_4094, MPWM1_PRDV=0x4000_5094

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		PERIOD_V	
-		0x0002	
-		RW	

15 0	PERIOD_V	16-bit PWM period The bit value must be larger than 0x0010.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the period for phase V only.

14.3.5 MPWMn_PRDW: MPWM n phase W period register

MPWMn_PRDW is a 32-bit register.

MPWM0_PRDW=0x4000_4098, MPWM1_PRDW=0x4000_5098

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		PERIOD_W	
-		0x0002	
-		RW	

15 0	PERIOD_W	16-bit PWM period The bit value must be larger than 0x0010.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the period for phase W only.

14.3.6 MPWMn_CNTU: MPWM n phase U period register

MPWMn_CNTU is a 32-bit register.

MPWM0_CNTU=0x4000_40A0, MPWM1_CNTU=0x4000_50A0

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	PWM counter value	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the value of the phase U counter.

14.3.7 MPWMn_CNTV: MPWM n phase V period register

MPWMn_CNTV is a 32-bit register.

MPWM0_CNTV=0x4000_40A4, MPWM1_CNTV=0x4000_50A4

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	PWM counter value	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the value of the phase V counter.

14.3.8 MPWMn_CNTW: MPWM n phase W period register

MPWMn_CNTW is a 32-bit register.

MPWM0_CNTW=0x4000_40A8, MPWM1_CNTW=0x4000_50A8

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	PWM counter value	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the value of the phase W counter.

14.3.9 MPWMn_DTRU: MPWM n phase U dead time register

MPWMn_DTRU is a 32-bit register. Its settings are applied to phases U, V, and W simultaneously in motor mode, and only to phase U in Individual mode.

MPWM0_DTRU=0x4000_40B0, MPWM1_DTRU=0x4000_50B0

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
DTMDSEL	Reserved	UDTEN	UPSHRT	Reserved	UDTCLK	UHDT	ULDT
0	-	0	0	-	00	0x00	0x00
RW	-	RW	RW	-	RW	RW	RW

31	DTMDSEL	Dead time mode selection When the POL function is used, a phase's H and L signals can overlap in the high level. This bit is typically set to prevent such a phenomenon. 0: Inserts dead time at the leading edge of PWMxH and the trailing edge of PWMxL 1: Inserts dead time at the trailing edge of PWMxH and the leading edge of PWMxL	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the dead time mode for phase U.
23	UDTEN	Whether to enable or disable the dead time function You must disable this function in two-channel symmetric mode because this mode does not support the function. 0: Disables the dead time function. 1: Enables the dead time function.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables the dead time function for phase U.
22	UPSHRT	Short-circuit protection This function is supported in two-channel symmetric mode. In one-channel mode, both the H and L sides are not simultaneously active. When either side is active, the other side always inactive. 0: Enables short-circuit protection (When the H- and L-side outputs are active simultaneously, they are all turned off.) 1: Disables short-circuit protection	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables short-circuit protection for phase U.
17 16	UDTCLK	Dead time prescaler 00: Sets PWM CLK/2 as the dead time counter clock. 01: Sets PWM CLK/4 as the dead time counter clock. 10: Sets PWM CLK/8 as the dead time counter clock. 11: Sets PWM CLK/16 as the dead time counter clock.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the dead time counter for phase U.
15 8	UHDT	Rising dead time value (set as the delay time from the normal polarity to the rising level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the dead time.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the falling dead time value for phase U.
7 0	ULDT	Falling dead time value (set as the delay time from the normal polarity to the falling level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the dead	

	time.	
	Motor mode	The bit is not used in this mode.
	Individual mode	The bit sets the falling dead time value for phase U.

NOTES:

1. This short-circuit protection is applied to the PWM’s internal signals and not to its external signals. When the internal signals from the H and L sides are all in the high level, short-circuit protection induces both sides to generate low-level outputs.
2. In Individual mode, setting only either one of UHDT and ULDT to zero can cause an abnormal operation. Therefore, both bit fields must all be set either to zero or to values other than zero.

14.3.10 MPWMn_DTRV: MPWM n phase V dead time register

MPWMn_DTRV is a 32-bit register. Its settings are not used in motor mode and applied only to phase V in Individual mode.

MPWM0_DTRV=0x4000_40B4, MPWM1_DTRV=0x4000_50B4

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
DTMDSEL	Reserved	VDTEN	VPSHRT	Reserved	VDTCLK	VHDT	VLDT
0	-	0	0	-	00	0x00	0x00
RW	-	RW	RW	-	RW	RW	RW

31	DTMDSEL	Dead time mode selection When the POL function is used, a phase’s H and L signals can overlap in the high level. This bit is typically set to prevent such a phenomenon. 0: Inserts dead time at the leading edge of PWMxH and the trailing edge of PWMxL 1: Inserts dead time at the trailing edge of PWMxH and the leading edge of PWMxL	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the dead time mode for phase V.
23	VDTEN	Whether to enable or disable the dead time function You must disable this function in two-channel symmetric mode because this mode does not support the function. 0: Disables the dead time function. 1: Enables the dead time function.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables the dead time function for phase V.
22	VPSHRT	Short-circuit protection This function is supported in two-channel symmetric mode. In one-channel mode, both the H and L sides are not simultaneously active. When either side is active, the other side always inactive. 0: Enables short-circuit protection (When the H- and L-side outputs are active simultaneously, they are all turned off.) 1: Disables short-circuit protection	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables short-circuit protection for phase V.
17 16	VDTCLK	Dead time prescaler 00: Sets PWM CLK/2 as the dead time counter clock. 01: Sets PWM CLK/4 as the dead time counter clock. 10: Sets PWM CLK/8 as the dead time counter clock. 11: Sets PWM CLK/16 as the dead time counter clock.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the dead time counter for phase V.
15 8	VHDT	Rising dead time value (set as the delay time from the normal polarity to the rising level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the dead time.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the falling dead time value for phase V.
7 0	VLDT	Falling dead time value (set as the delay time from the normal polarity to the falling level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the	

	dead time.	
	Motor mode	The bit is not used in this mode.
	Individual mode	The bit sets the falling dead time value for phase V.

NOTES:

1. This short-circuit protection is applied to the PWM’s internal signals and not to its external signals. When the internal signals from the H and L sides are all in the high level, short-circuit protection induces both sides to generate low-level outputs.
2. In Individual mode, setting only either one of VHDT and VLDT to zero can cause an abnormal operation. Therefore, both bit fields must all be set either to zero or to values other than zero.

14.3.11 MPWMn_DTRW: MPWM n phase W dead time register

MPWMn_DTRW is a 32-bit register. Its settings are not used in motor mode and applied only to phase W in Individual mode.

MPWM0_DTRW=0x4000_40B8, MPWM1_DTRW=0x4000_50B8

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
DTMDSEL	Reserved	WDTEN	WPSHRT	Reserved	WDTCLK	WHDT	WLDT
0	-	0	0	-	00	0x00	0x00
RW	-	RW	RW	-	RW	RW	RW

31	DTMDSEL	Dead time mode selection When the POL function is used, a phase's H and L signals can overlap in the high level. This bit is typically set to prevent such a phenomenon. 0: Inserts dead time at the leading edge of PWMxH and the trailing edge of PWMxL 1: Inserts dead time at the trailing edge of PWMxH and the leading edge of PWMxL	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit selects the dead time mode for phase W.
23	WDTEN	Whether to enable or disable the dead time function You must disable this function in two-channel symmetric mode because this mode does not support the function. 0: Disables the dead time function. 1: Enables the dead time function.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables the dead time function for phase W.
22	WPSHRT	Short-circuit protection This function is supported in two-channel symmetric mode. In one-channel mode, both the H and L sides are not simultaneously active. When either side is active, the other side always inactive. 0: Enables short-circuit protection (When the H- and L-side outputs are active simultaneously, they are all turned off.) 1: Disables short-circuit protection	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit enables short-circuit protection for phase W.
17 16	WDTCLK	Dead time prescaler 00: Sets PWM CLK/2 as the dead time counter clock. 01: Sets PWM CLK/4 as the dead time counter clock. 10: Sets PWM CLK/8 as the dead time counter clock. 11: Sets PWM CLK/16 as the dead time counter clock.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the dead time counter for phase W.
15 8	WHDT	Rising dead time value (set as the delay time from the normal polarity to the rising level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the dead time.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit sets the falling dead time value for phase W.
7 0	WLDT	Falling dead time value (set as the delay time from the normal polarity to the falling level output) 0x01 through 0xFF: dead time **NOTE:** Writing a 0 to the bit will have the same effect as disabling the dead	

	time.	
	Motor mode	The bit is not used in this mode.
	Individual mode	The bit sets the falling dead time value for phase W.

NOTES:

1. This short-circuit protection is applied to the PWM’s internal signals and not to its external signals. When the internal signals from the H and L sides are all in the high level, short-circuit protection induces both sides to generate low-level outputs.
2. In Individual mode, setting only either one of WHDT and WLDT to zero can cause an abnormal operation. Therefore, both bit fields must all be set either to zero or to values other than zero.

14.3.12 MPWMn_CAPCNTx: MPWM n phase U/V/W capture counter register

MPWMn_CAPCNTx is a 32-bit register. It displays a 17-bit capture counter value made with the main counter value. In Individual mode, phases U, V, and W have different capture counters. The register can only be used in Individual mode and not in motor mode.

MPWM0_CAPCNTU=0x4000_40C0, MPWM0_CAPCNTV=0x4000_40C4, MPWM0_CAPCNTW=0x4000_40C8, MPWM1_CAPCNTU=0x4000_50C0, MPWM1_CAPCNTV=0x4000_50C4, MPWM1_CAPCNTW=0x4000_50C8

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
CNTCLEAR	Reserved			CAPEN	Reserved										CAPCNTx																
0	-			0	-										0x00001																
WO	-			RW	-										RO																

31	CNTCLEAR	Whether or not to clear the capture counter (Auto-clear) 0: No effect 1: Clears the counter.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit clears the capture counter value.
27	CAPEN	Whether to enable or disable the capture function 0: Disables the capture function. 1: Enables the capture function.	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit determines the enablement of the capture function.
16 0	CAPCNTx (x=U, V, W)	PWM capture counter value	
		Motor mode	The bit is not used in this mode.
		Individual mode	The bit displays the capture counter value for the corresponding phase.

NOTE: You can start the capture counter by enabling the main counter start bit. If the main counter is an up-counter, only 16 bits are used in the register.

14.3.13 MPWMn_RCAPx: MPWM n phase U/V/W capture rising value register

MPWMn_RCAPx is a 32-bit register that imports the capture counter value when the external signal is rising. It can be used in Individual mode but not in motor mode.

MPWM0_RCAPU=0x4000_40D0, MPWM0_RCAPV=0x4000_40D4, MPWM0_RCAPW=0x4000_40D8
MPWM1_RCAPU=0x4000_50D0, MPWM1_RCAPV=0x4000_50D4, MPWM1_RCAPW=0x4000_50D8

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RCAPFLAG</td><td colspan="14">Reserved</td><td colspan="17">RCAPx</td></tr>
<tr><td>0</td><td colspan="14">-</td><td colspan="17">0x00000</td></tr>
<tr><td>RC</td><td colspan="14">-</td><td colspan="17">RO</td></tr>
</table>

31	RCAPFLAG	Capture counter flag (Auto-clear) 0: No capture value has been imported. 1: The capture value has been received (Writing a 1 to the bit clears RCAPx and the flag).	
		Motor mode	The bit is not used in this mode.
		Individual mode	Writing to the bit clears the capture value.
16 0	RCAPx (x=U, V, W)	Capture counter value when the external signal is rising	
		Motor mode	The bit is not used in this mode.
		Individual mode	Capture counter value when the external signal is rising

14.3.14 MPWMn_FCAPx: MPWM n phase U/V/W capture falling value register

MPWMn_FCAPx is a 32-bit register that imports the capture counter value when the external signal is falling. It can be used in Individual mode but not in motor mode.

MPWM0_FCAPU=0x4000_40E0, MPWM0_FCAPV=0x4000_40E4, MPWM0_FCAPW=0x4000_40E8
MPWM1_FCAPU=0x4000_50E0, MPWM1_FCAPV=0x4000_50E4, MPWM1_FCAPW=0x4000_50E8

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>FCAPFLAG</td><td colspan="14">Reserved</td><td colspan="17">FCAPx</td></tr>
<tr><td>0</td><td colspan="14">-</td><td colspan="17">0x00000</td></tr>
<tr><td>RC</td><td colspan="14">-</td><td colspan="17">RO</td></tr>
</table>

31	FCAPFLAG	Capture counter flag (Auto-clear) 0: No capture value has been imported. 1: The capture value has been received (Writing a 1 to the bit clears FCAPx and the flag).	
		Motor mode	The bit is not used in this mode.
		Individual mode	Writing to the bit clears the capture value.
16 0	FCAPx (x=U, V, W)	Capture counter value when the external signal is falling	
		Motor mode	The bit is not used in this mode.
		Individual mode	Capture counter value when the external signal is falling

14.3.15 MPWMn_SCAPx: MPWM n phase U/V/W sub-capture value register

MPWMn_SCAPx is a 32-bit register that imports the capture counter value when the external signal is rising or falling. It can be used in Individual mode but not in motor mode.

MPWM0_SCAPU=0x4000_40F0, MPWM0_SCAPV=0x4000_40F4, MPWM0_SCAPW=0x4000_40F8
MPWM1_SCAPU=0x4000_50F0, MPWM1_SCAPV=0x4000_50F4, MPWM1_SCAPW=0x4000_50F8

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
SCAPFLAG	Reserved	DEGESEL	Reserved	SCAPx
0	-	0	-	0x00000
RW	-	RW	-	RO

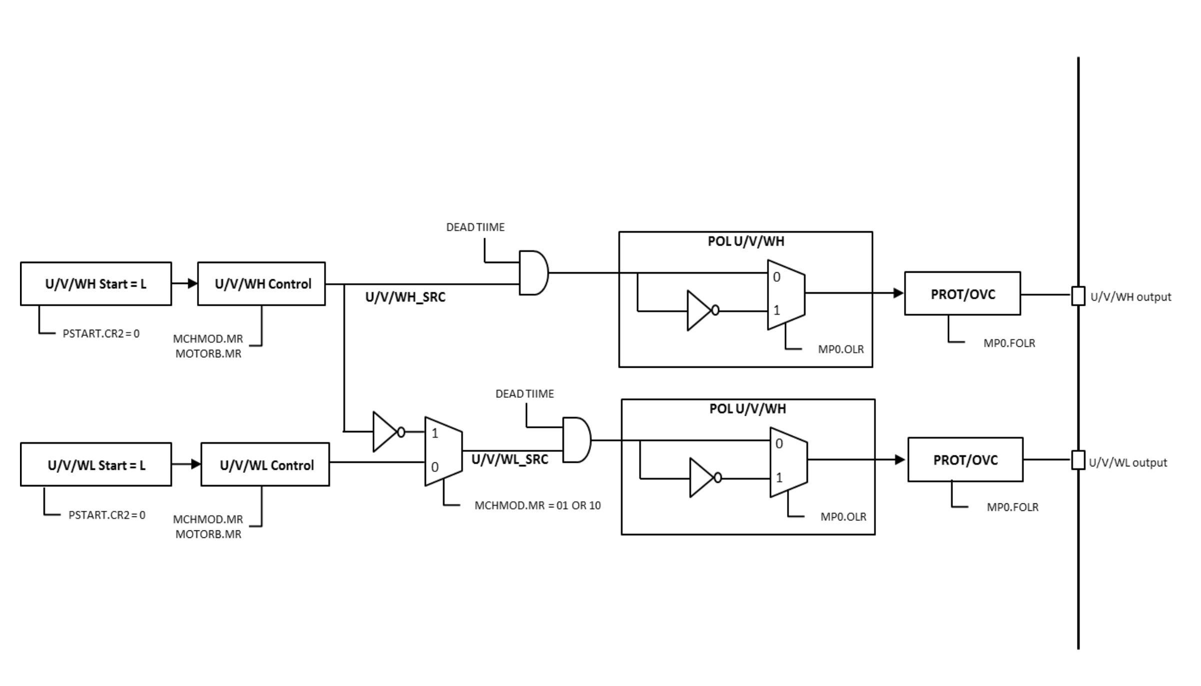
31	SCAPFLAG	Sub-capture counter flag (Auto-clear) 0: No capture value has been imported. 1: The capture value has been received (Writing a 1 to the bit clears SCAPx and the flag).	
		Motor mode	The bit is not used in this mode.
		Individual mode	Writing to the bit clears the capture value.
28	EDGESEL	Selection of the external signal's edge at which the counter value is captured 0: Captured at the rising edge 1: Captured at the falling edge	
		Motor mode	The bit is not used in this mode.
		Individual mode	The counter value is captured at the selected edge.
16 0	SCAPx (x=U, V, W)	Capture counter value when the external signal is rising/falling	
		Motor mode	External signal capture counter value
		Individual mode	The bit is not used in this mode.

14.4 Functional description

The MPWM includes three channels, each of which controls a pair of outputs. In normal PWM mode, each channel runs independently. Thus, a total of six PWM outputs can be generated.

Each PWM output is built with various settings. The figure below shows the process of generating PWM output signals.

Figure 99. PWM Output Generation Chain



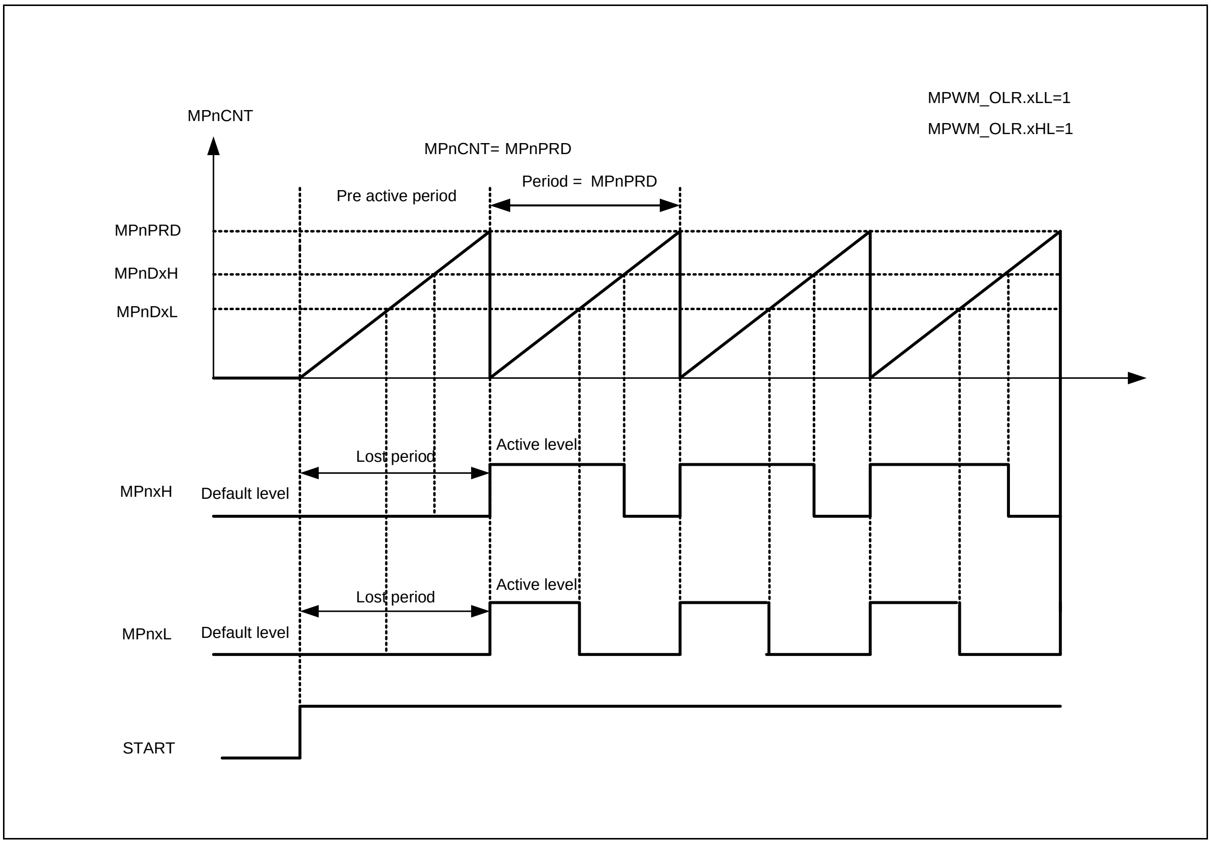
14.4.1 Normal PWM up-count mode timing

In normal PWM mode, each channel runs independently. A total of six PWM outputs can be generated. The figure below shows an example of waveforms generated in this mode. The PWM outputs are maintained in low state by default until PSTART is enabled. When START is enabled, the period counter starts counting up to the PRD count value. In the first period, the MPWM does not generate PWM pulses.

Their generation starts from the second period. They are at their active levels during the duty value from the start of the counter value.

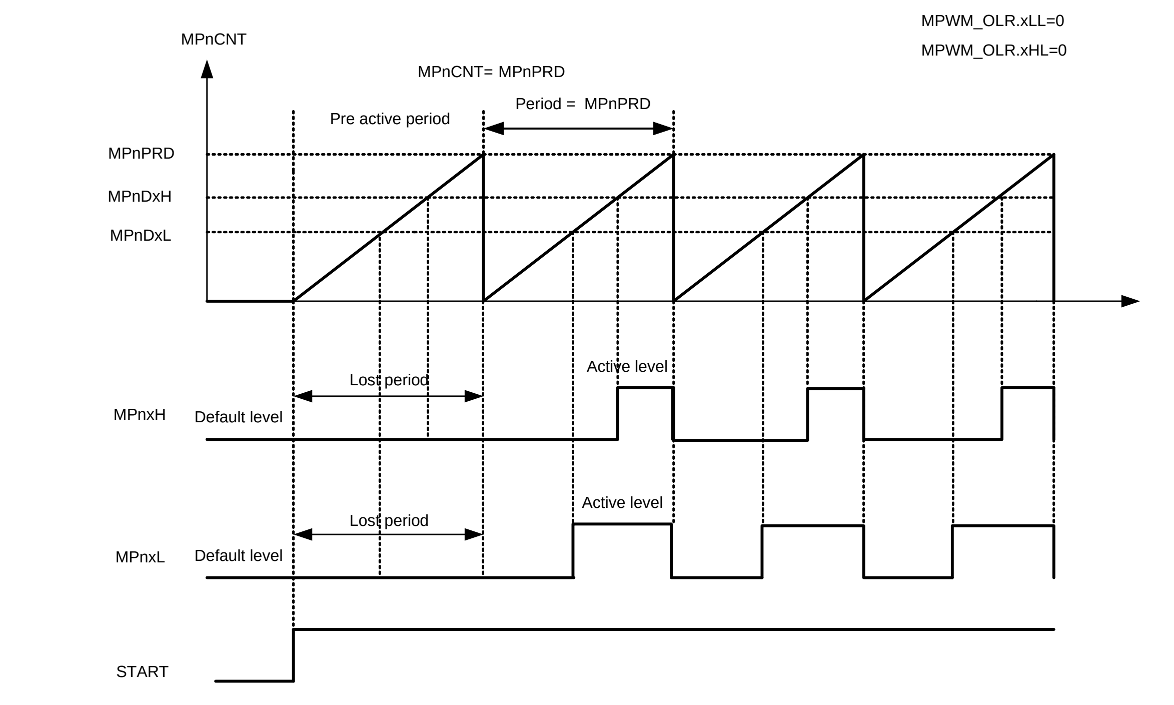
The figure below shows the waveform when the xHL, XLL PWM output of MPWM_OLR is set to "High".

Figure 100. Up-count Mode Waveforms: MOTORB = 1, UPDOWN = 0, OLR.xLL =1, OLR.xHL =1



The figure below shows the waveform when the xHL, XLL PWM output of MPWM_OLR is set to "Low".

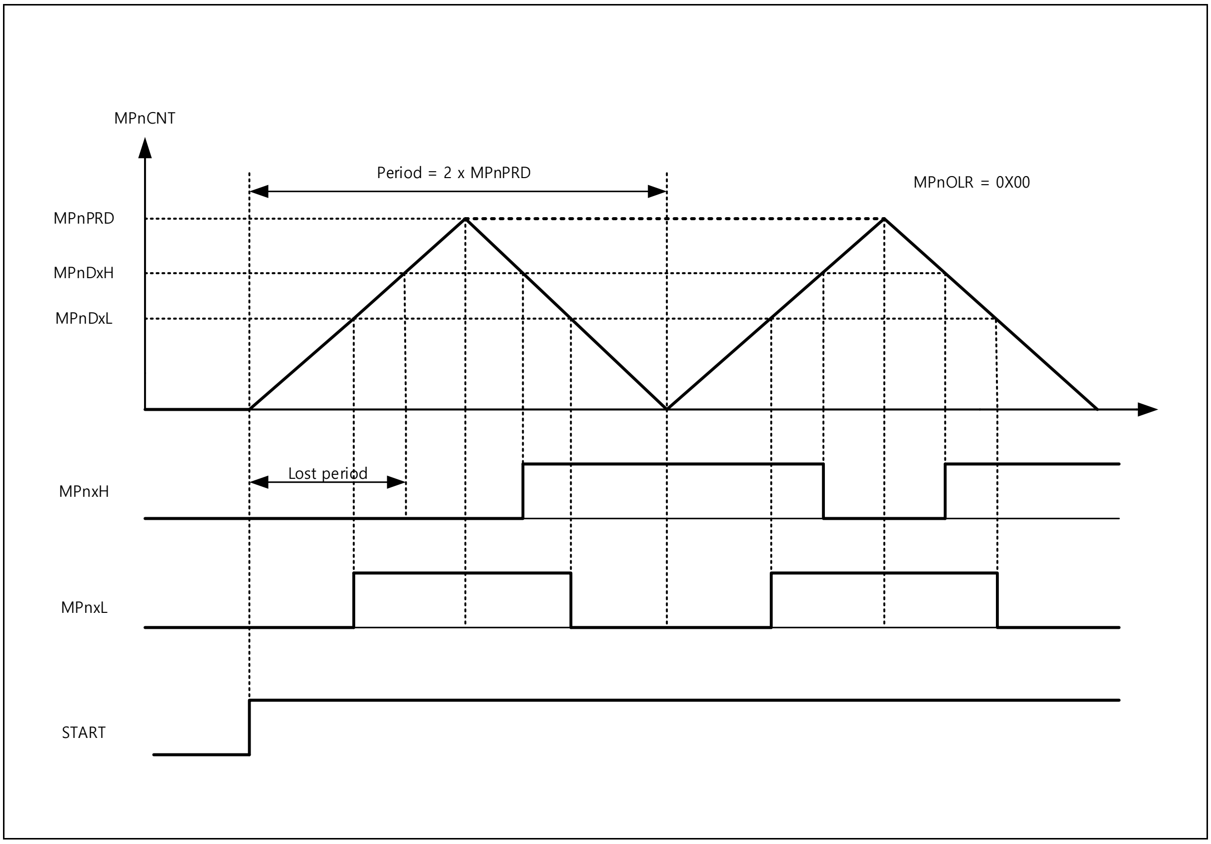
Figure 101. Up-count Mode Waveforms: MOTORB =1, UPDOWN =0, OLR.xLL =0, OLR.xHL =0



14.4.2 Normal PWM up-down count mode timing

Basic operations are identical in up-down count mode and the up-count mode. Their difference is that the former has a twice Individual period. The default active level is opposite in a pair PWM output. The output polarity is controlled in the OLR register.

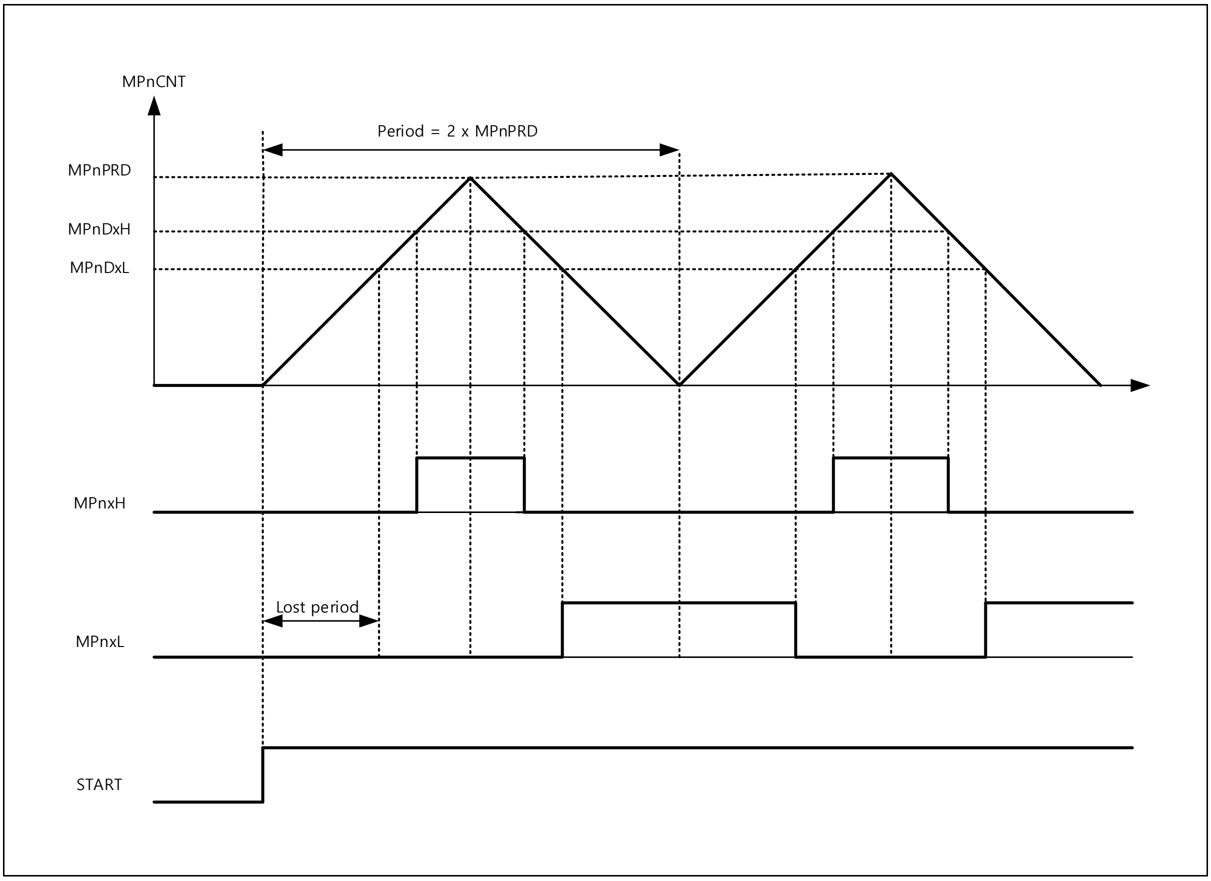
Figure 102. Up-down Count Mode Waveforms: MOTORB = 0, MCHMOD = 0, UPDOWN = 1



14.4.3 MPWM two-channel symmetric mode timing

There are three different modes for motor PWM operations. Two-channel symmetric mode, one-channel symmetric mode, and one-channel asymmetric mode. The figure below shows waveforms in two-channel symmetric mode.

Figure 103. Two-channel Asymmetric Mode Waveforms: MOTORB = 0, MCHMOD = 00



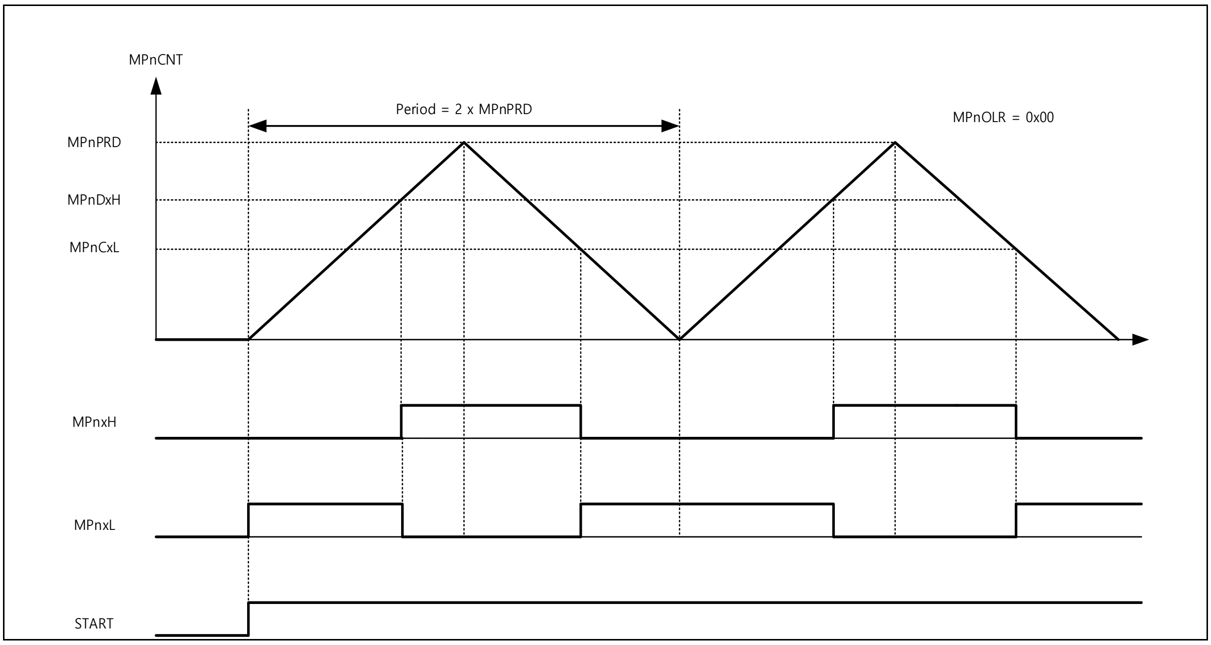
Both the H and L sides start with a “low” level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the duty level in the up-count period. It changes back to the default level when the counter matches the duty level in the down-count period.

The symmetrical function is controlled by the corresponding duty register value of each channel.

14.4.4 Motor PWM one-channel asymmetric mode timing

In one-channel asymmetric mode, asymmetric pulses are created based on the settings in the H- and L-side duty registers. The L-side signal is always the inverse of the H-side signal. During up-count periods, the H-side signal's duty register matches create active level pulses. And during down-count periods, the L-side signal's duty register matches create default level pulses.

Figure 104. One-channel Asymmetric Mode Waveforms: MOTORB = 0, MCHMOD = 01



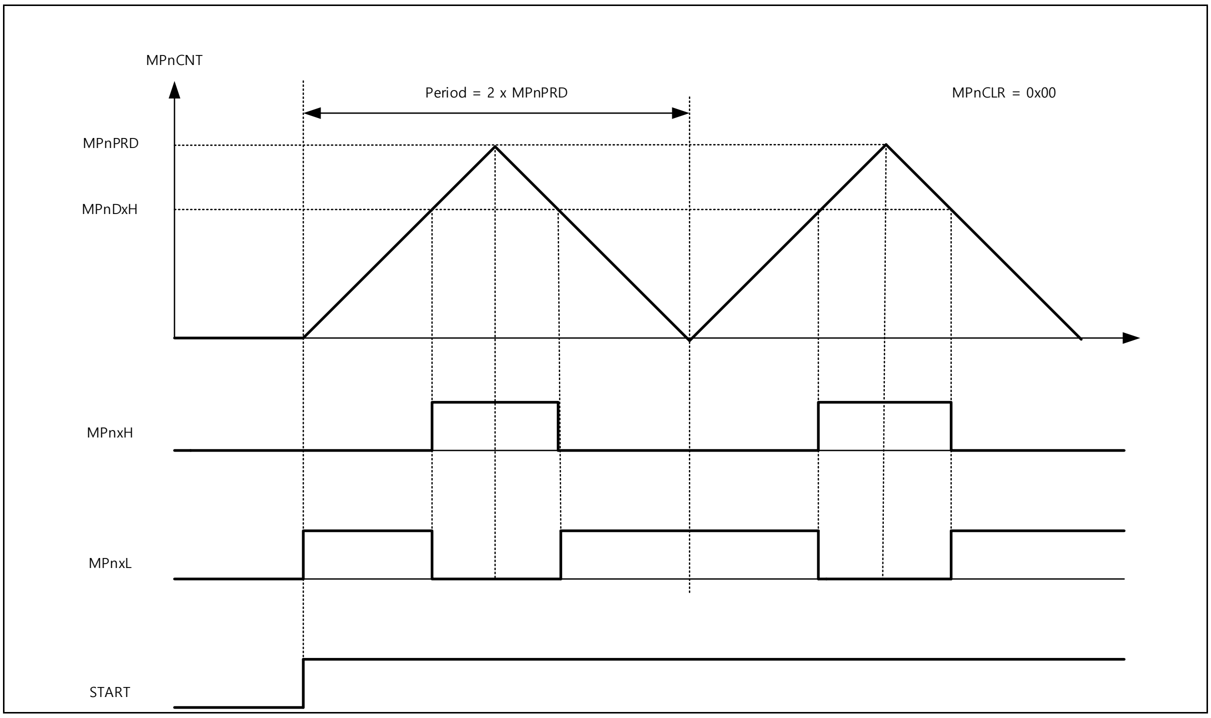
Both the H and L sides start with a "low" level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the H-side duty level in the up-count period. It changes back to the default level when the counter matches the L-side duty level in the down-count period.

When START becomes active, the L-side PWM output is converted to the active level, and the H-side outputs an inverted signal.

14.4.5 Motor PWM one-channel symmetric mode timing

In one-channel symmetric mode, symmetric period pulses are based on the settings of the H-side duty register. The L-side signal is always the inverse of the H-side signal. During up-count periods, the H-side signal's duty register matches create active level pulses. And during down-count periods, the H-side signal's duty register matches create default level pulses.

Figure 105. One-channel Symmetric Mode Waveforms: MOTORB = 0, MCHMOD = 10



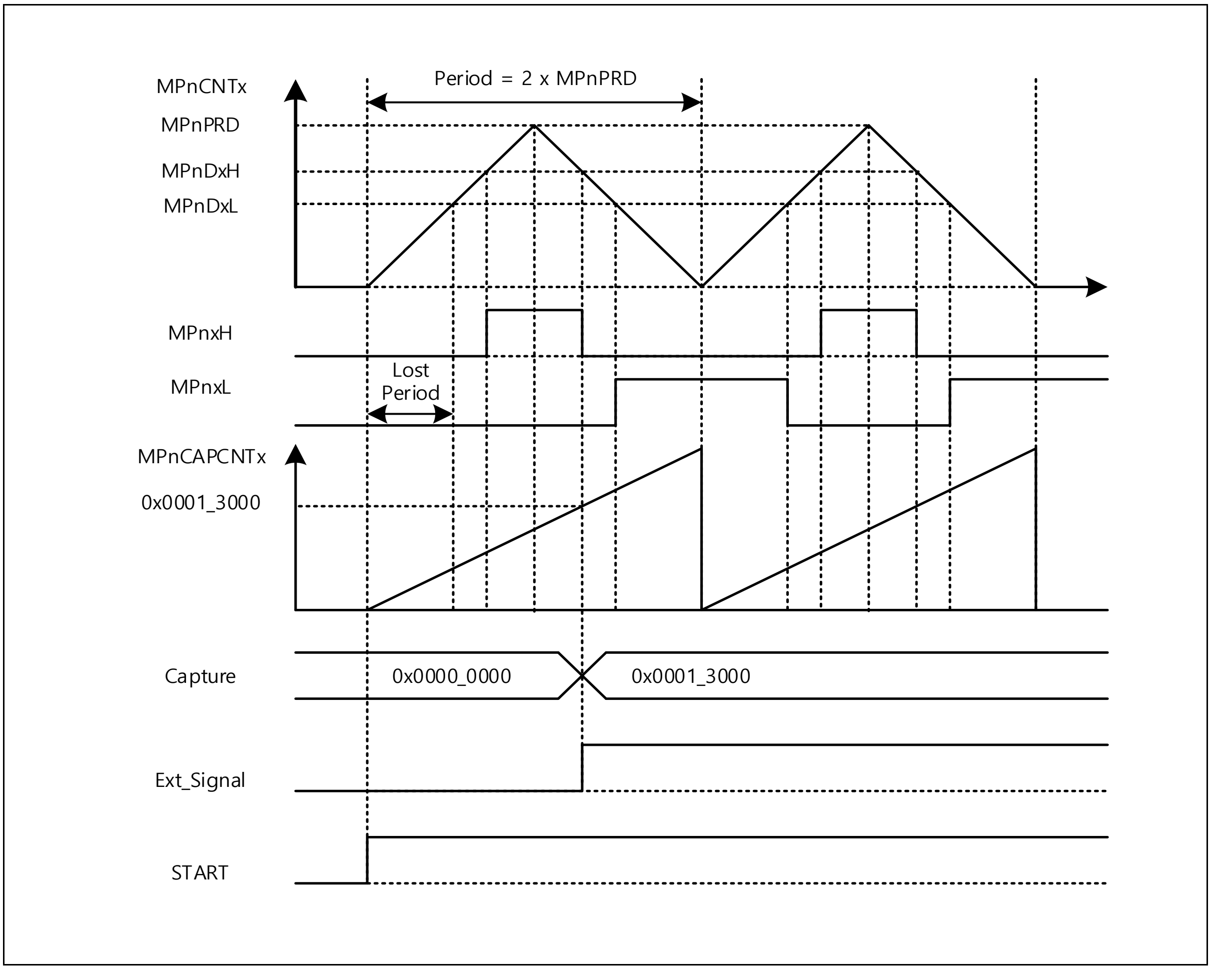
Both the H and L sides start with a "low" level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the H-side duty level in the up-count period. It changes back to the default level when the counter matches the H-side duty level in the down-count period.

When START becomes active, the L-side PWM output is converted to the active level, and the H-side outputs an inverted signal.

14.4.6 Individual PWM two-channel symmetric mode timing

Like motor PWM operations, there are three different modes for Individual PWM operations: two-channel symmetric mode, one-channel symmetric mode, and one-channel asymmetric mode. The figure below shows waveforms in two-channel symmetric mode.

Figure 106. Two-channel Asymmetric Mode Waveforms: MOTORB = 3, MCHMOD = 00



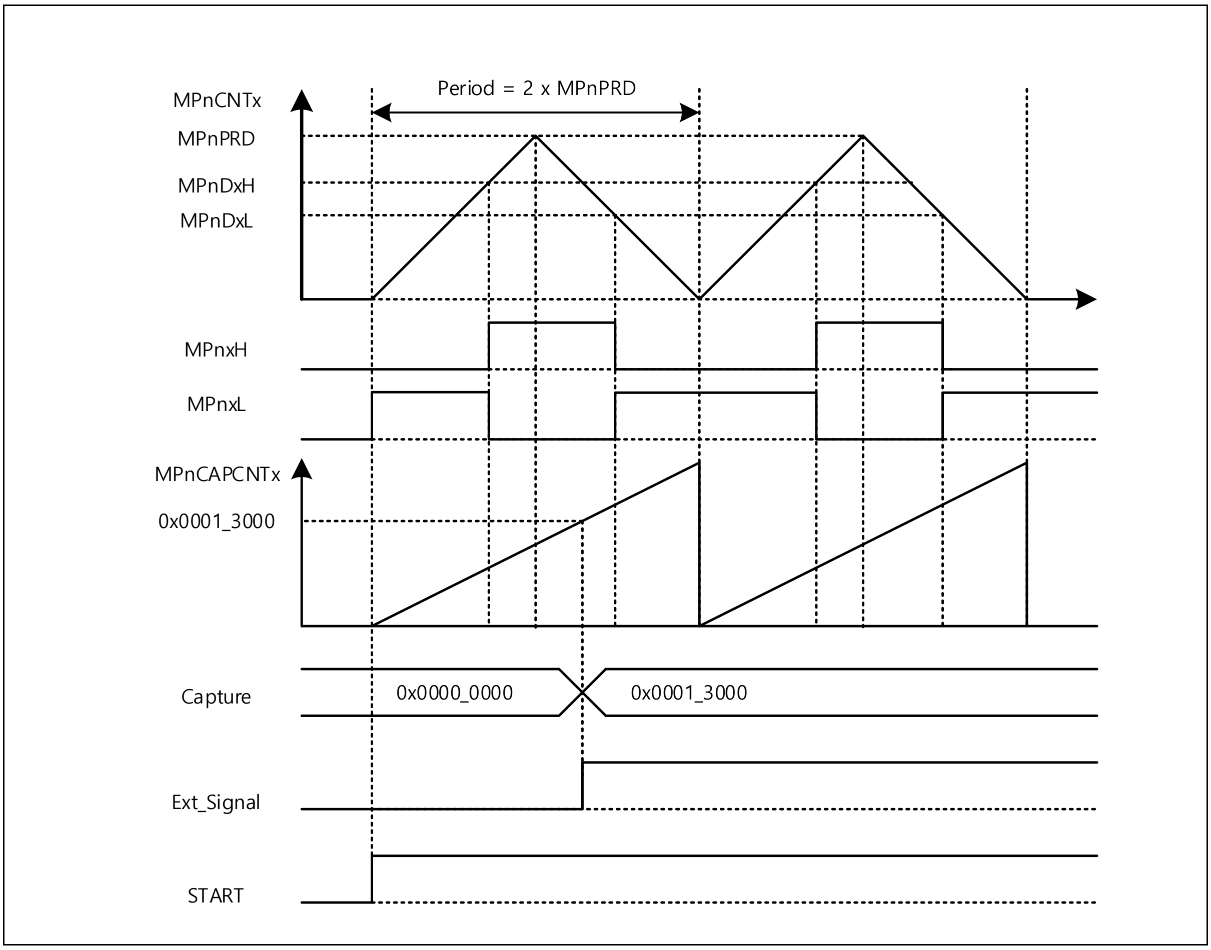
Both the H and L sides start with a “low” level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the duty level in the up-count period. It changes back to the default level when the counter matches the duty level in the down-count period.

The symmetrical function is controlled by the corresponding duty register value of each channel.

14.4.7 Individual PWM one-channel asymmetric mode timing

In one-channel asymmetric mode, asymmetric pulses are created based on the settings in the H- and L-side duty registers. The L-side signal is always the inverse of the H-side signal. During up-count periods, the H-side signal's duty register matches create active level pulses. And during down-count periods, the L-side signal's duty register matches create default level pulses.

Figure 107. One-channel Asymmetric Mode Waveforms: MOTORB = 3, MCHMOD = 01



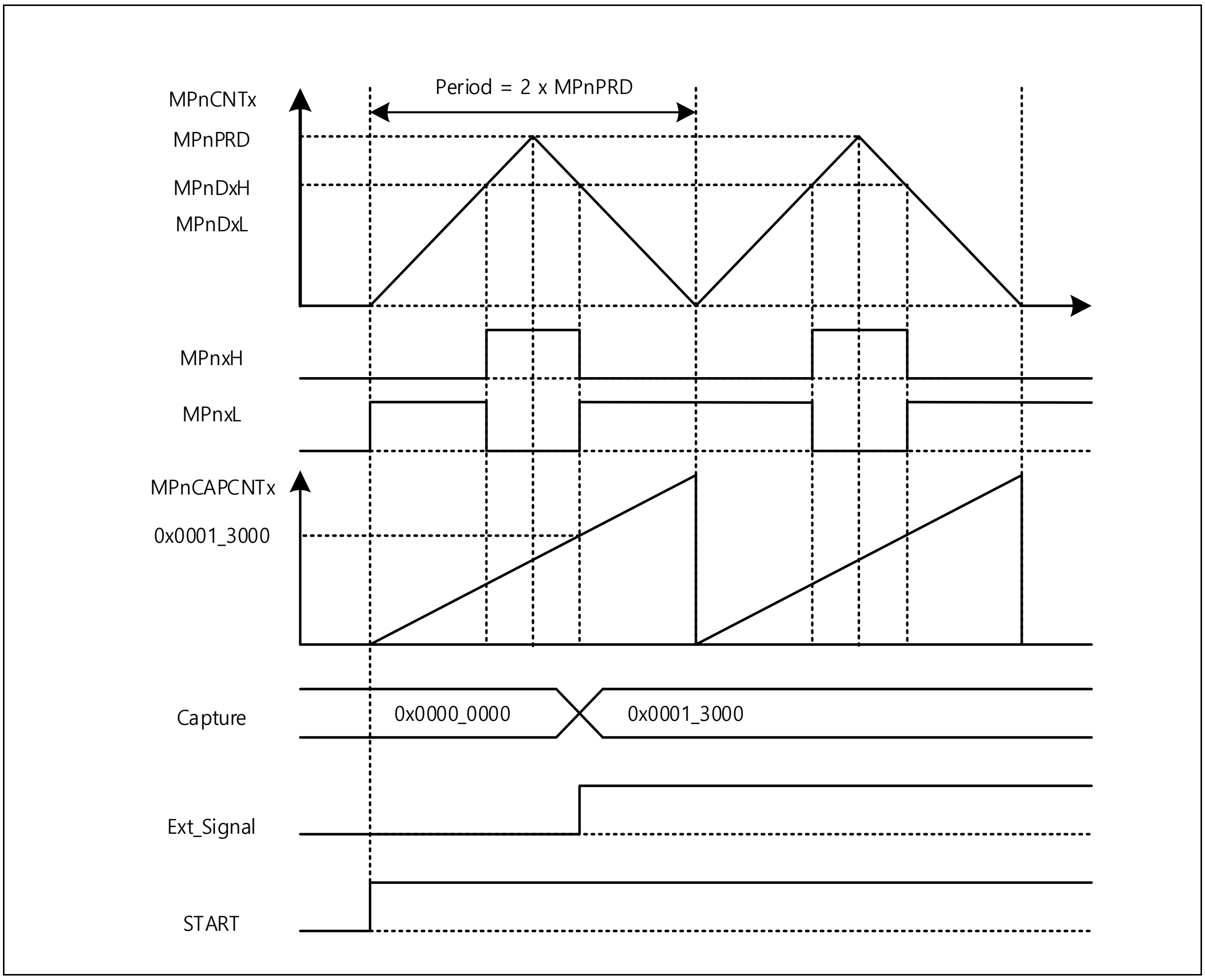
Both the H and L sides start with a "low" level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the H-side duty level in the up-count period. It changes back to the default level when the counter matches the L-side duty level in the down-count period.

When PSTART becomes active, the L-side PWM output is converted to the active level, and the H-side outputs an inverted signal.

14.4.8 Individual PWM one-channel symmetric mode timing

In one-channel symmetric mode, symmetric period pulses are based on the settings of the H-side duty register. The L-side signal is always the inverse of the H-side signal. During up-count periods, the H-side signal's duty register matches create active level pulses. And during down-count periods, the H-side signal's duty register matches create default level pulses.

Figure 108. One-channel Symmetric Mode Waveforms: MOTORB = 3, MCHMOD = 10



Both the H and L sides start with a “low” level signal by default. For the H side, the PWM output level changes to the active level when the PWM counter matches the H-side duty level in the up-count period. It changes back to the default level when the counter matches the H-side duty level in the down-count period.

When PSTART becomes active, the L-side PWM output is converted to the active level, and the H-side outputs an inverted signal.

14.4.9 MPWM dead time operation

To prevent external short-circuits, the MPWM provides a dead-time function. This function can be used only in motor mode and Individual mode. When either the H- or L-side output changes to the active level, the amount of dead time is inserted if the DTR.DTEN bit is enabled.

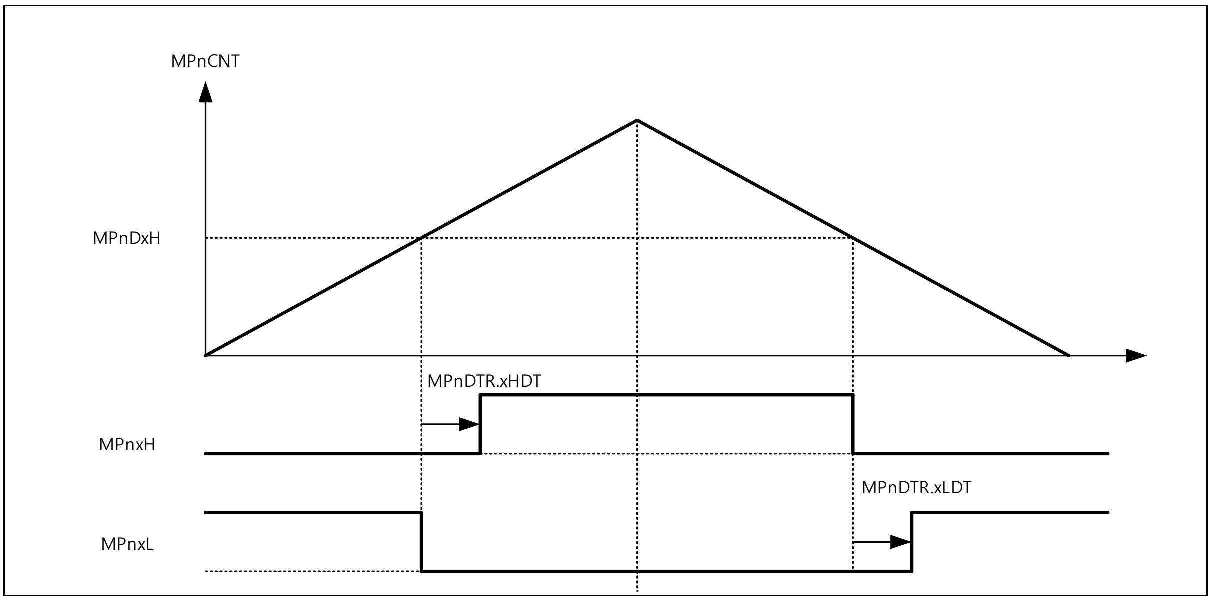
(The figure is an example of when the DTMDSEL bit is set to 0).

The dead time period is determined by the value of the value of DTR field.

When the PWM counter reaches the duty cycle value, the PWM output is masked and the dead time counter starts running. When the dead time counter reaches the value set in DT [7: 0], the output mask is set disabled.

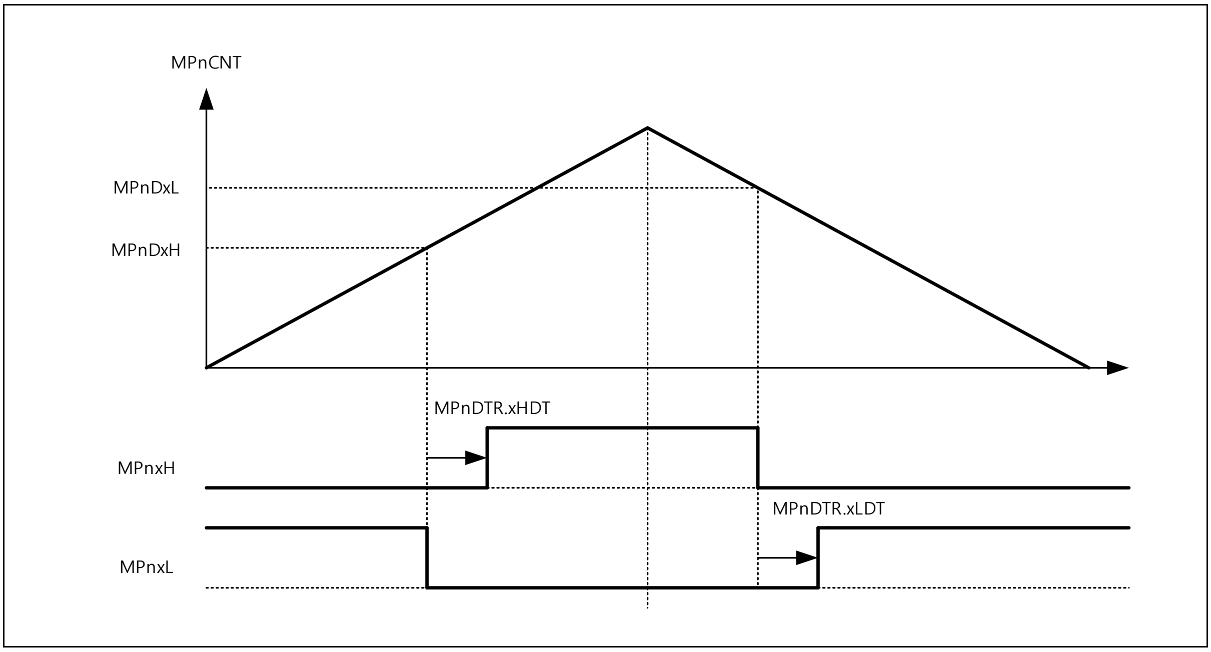
The following figure is an example of how dead time works in the one-channel symmetrical mode.

Figure 109. Dead Time Operation Timing Diagram: Symmetric Mode, DTMDSEL = 0



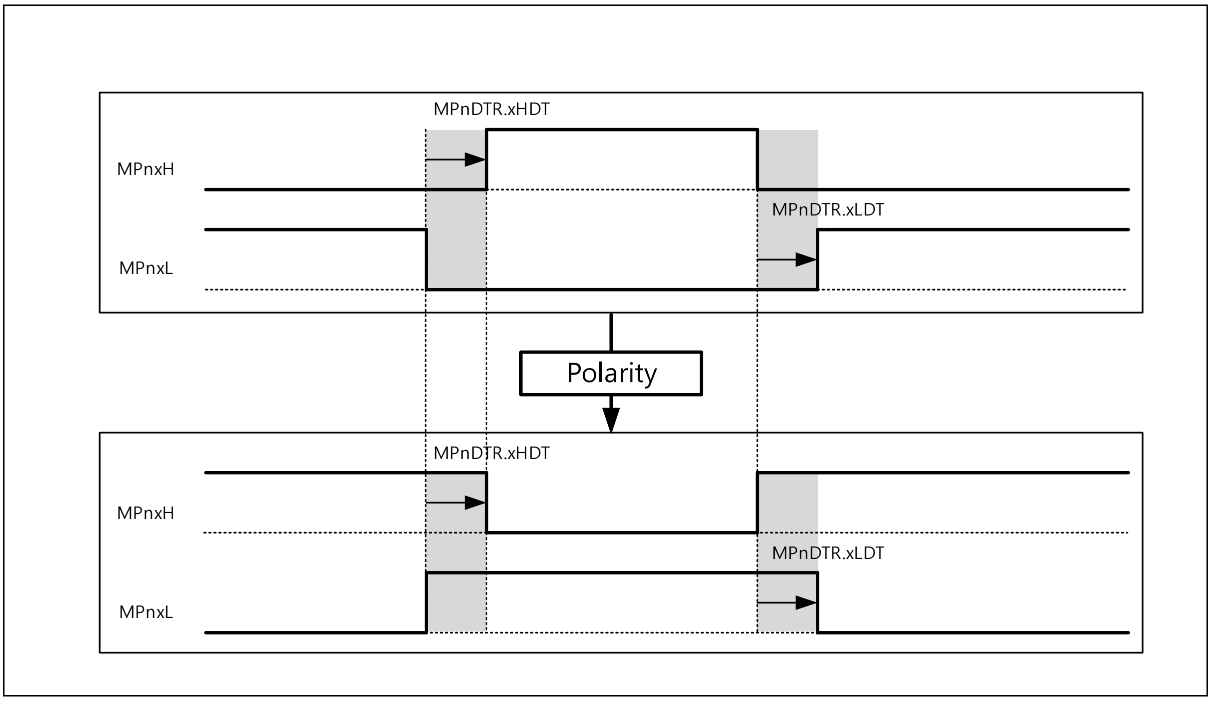
The figure below is an example of how dead time works in the one-channel asymmetric mode.

Figure 110. Dead Time Operation Timing Diagram: Asymmetric Mode, DTMDSEL = 0



If the polarity function is used, setting the dead time can cause both the H and L sides to become high. This can create problems when using highly active devices. In this case, you can set the DTMDSEL bit to 1 to change the point at which dead time is generated, as shown in Figure 112 and Figure 113.

Figure 111. Dead Time Polarity Settings



The following figure is based on the waveforms generated when polarity is applied.

Figure 112. Dead Time Operation Timing Diagram: Symmetric Mode, DTMDSEL = 1

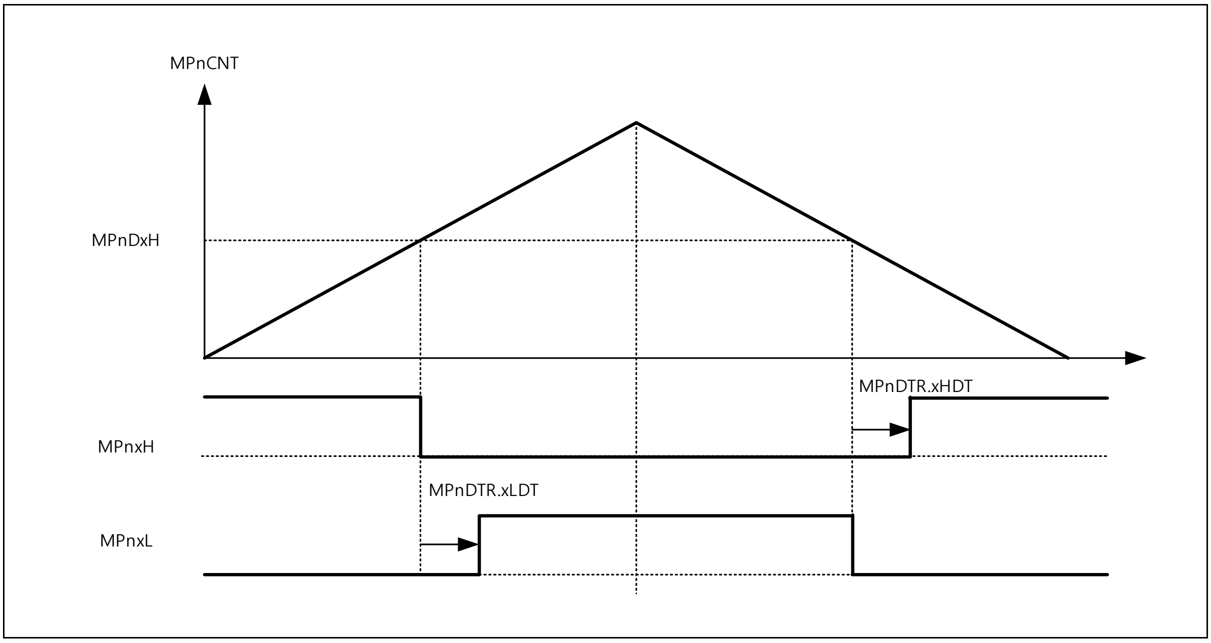
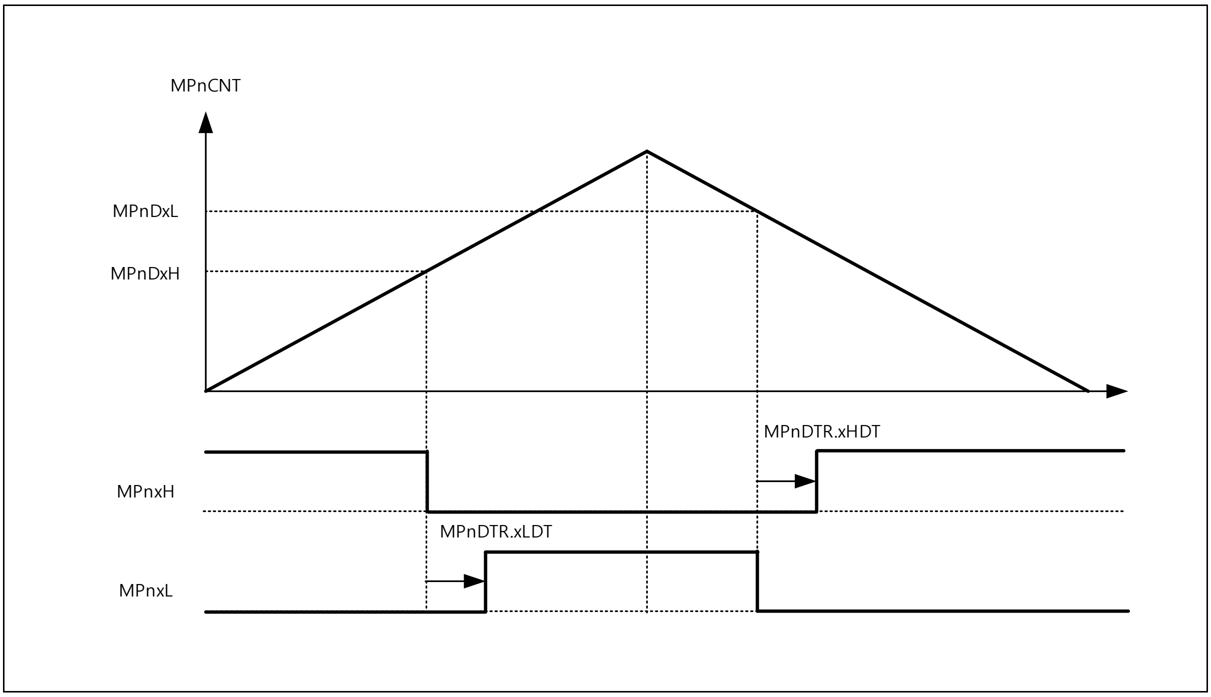


Figure 113. Dead Time Operation Timing Diagram: Asymmetric Mode, DTMDSEL = 1

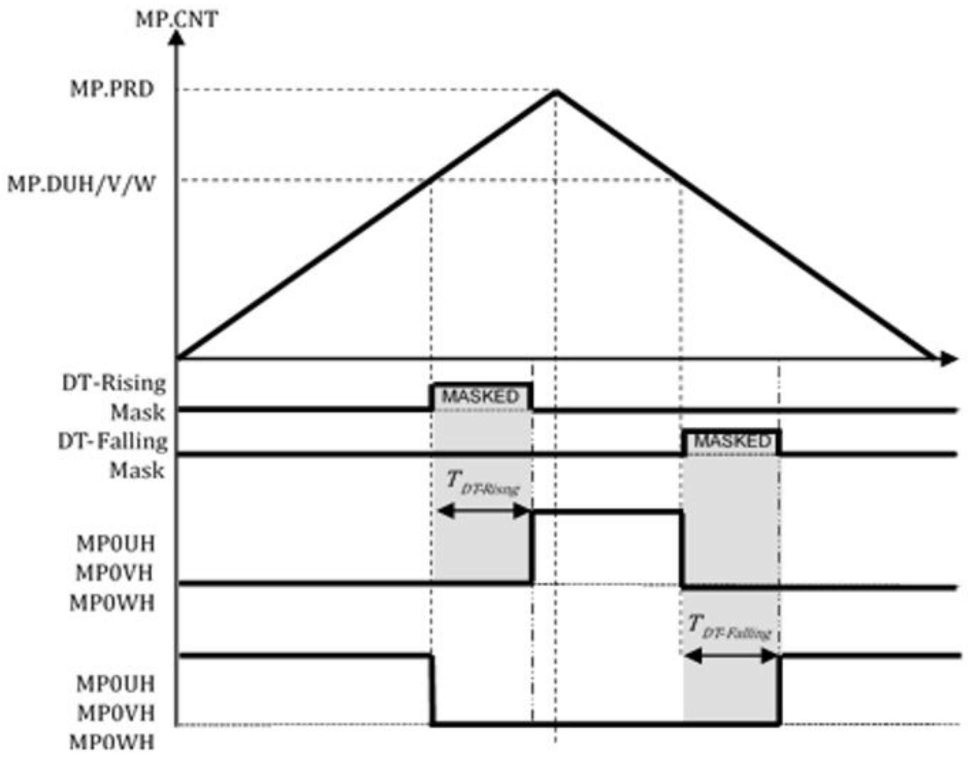


The dead time function is not available in two-channel symmetric mode. Dead time is inserted by the duty control of each channel.

14.4.10 Special case examples of MPWM dead time timing

The following figure shows how dead time works in typical situations. Dead time masking is enabled by running the duty match time and dead time counter. When the dead time counter reaches the dead time value, the mask is set disabled.

Figure 114. Typical Dead Time Operation: TDUTY>TDT



The figures below show some special case examples of dead time configurations:

Figure 115. H-side Minimum Pulse Timing: $T_{DUTY}<T_{DT}<2xT_{DUTY}$

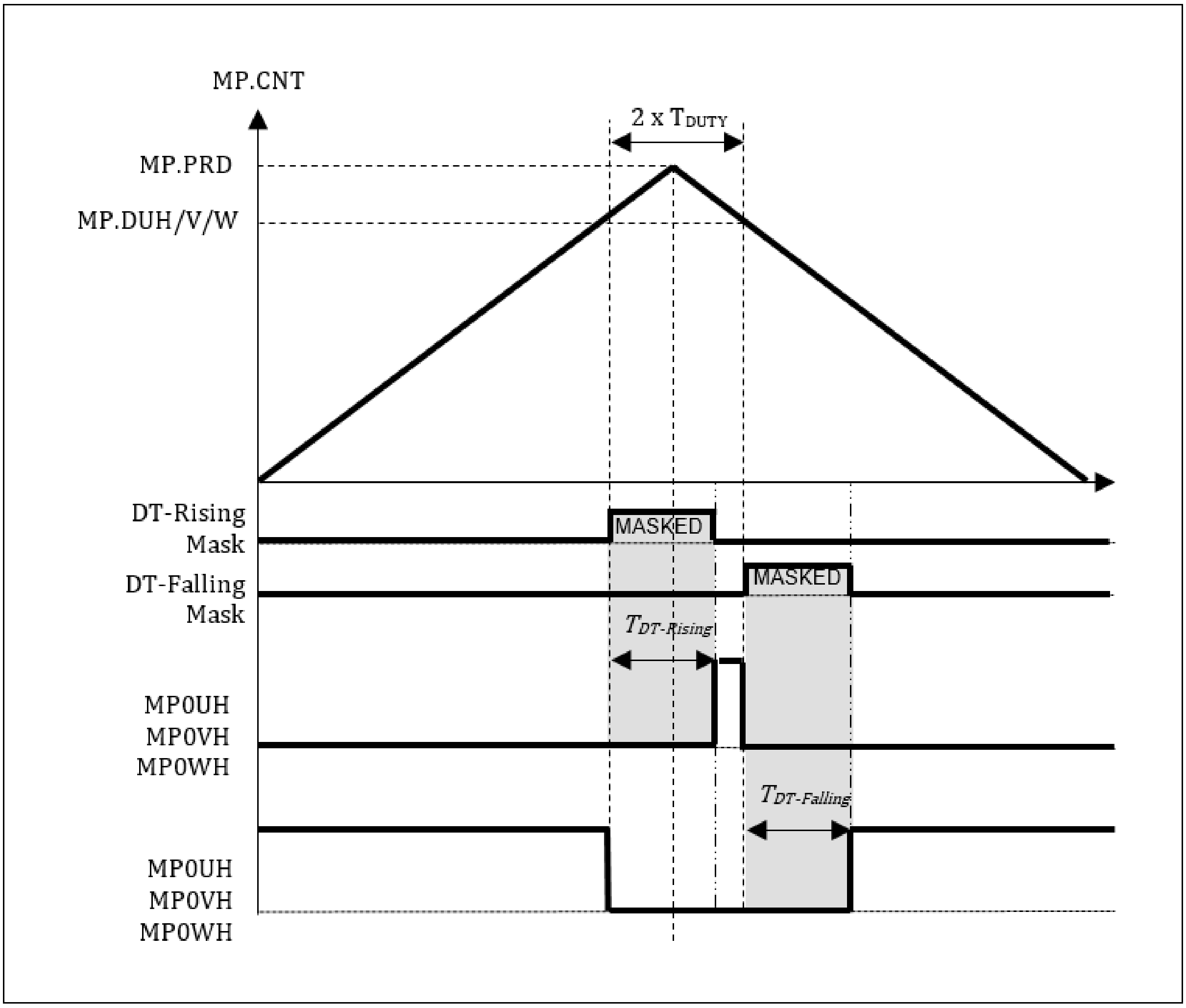


Figure 116. H-side Zero Pulse Timing: T_{DT}>2xT_{DUTY}

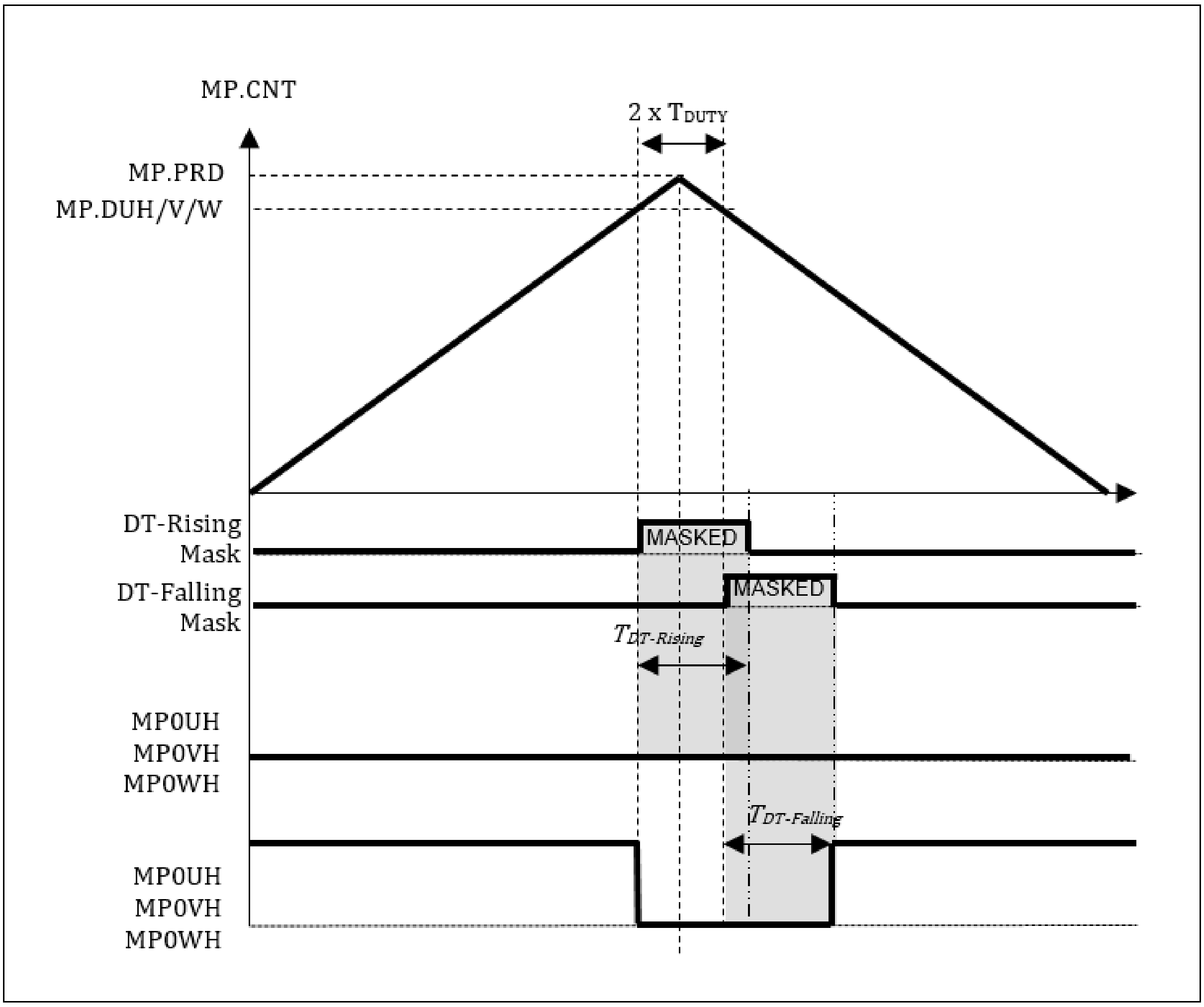


Figure 117. L-side Minimum Pulse Timing: T_{DT}<Period-T_{DUTY}

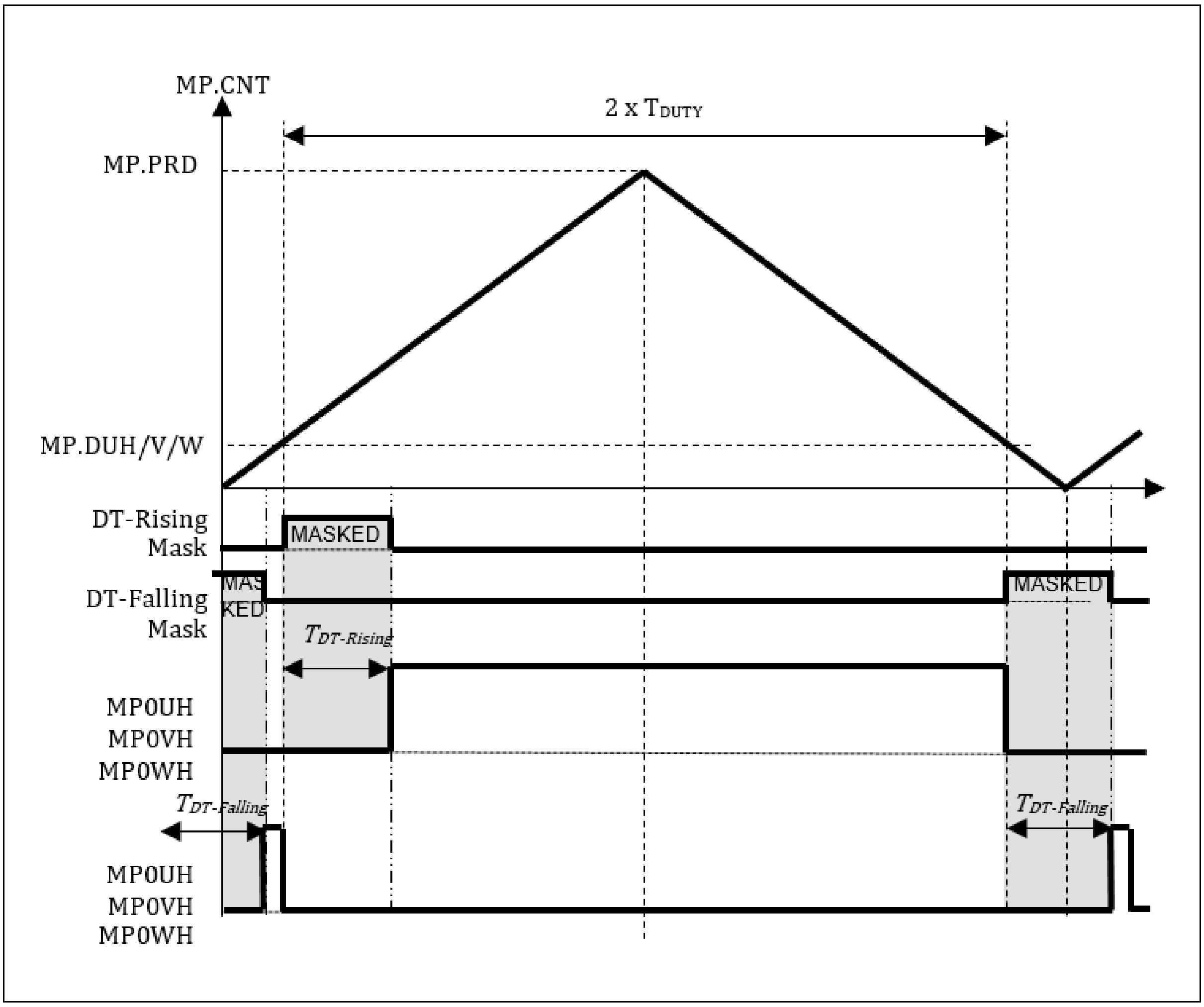


Figure 118. L-side Zero Pulse Timing: T_{DT}>Period-T_{DUTY}

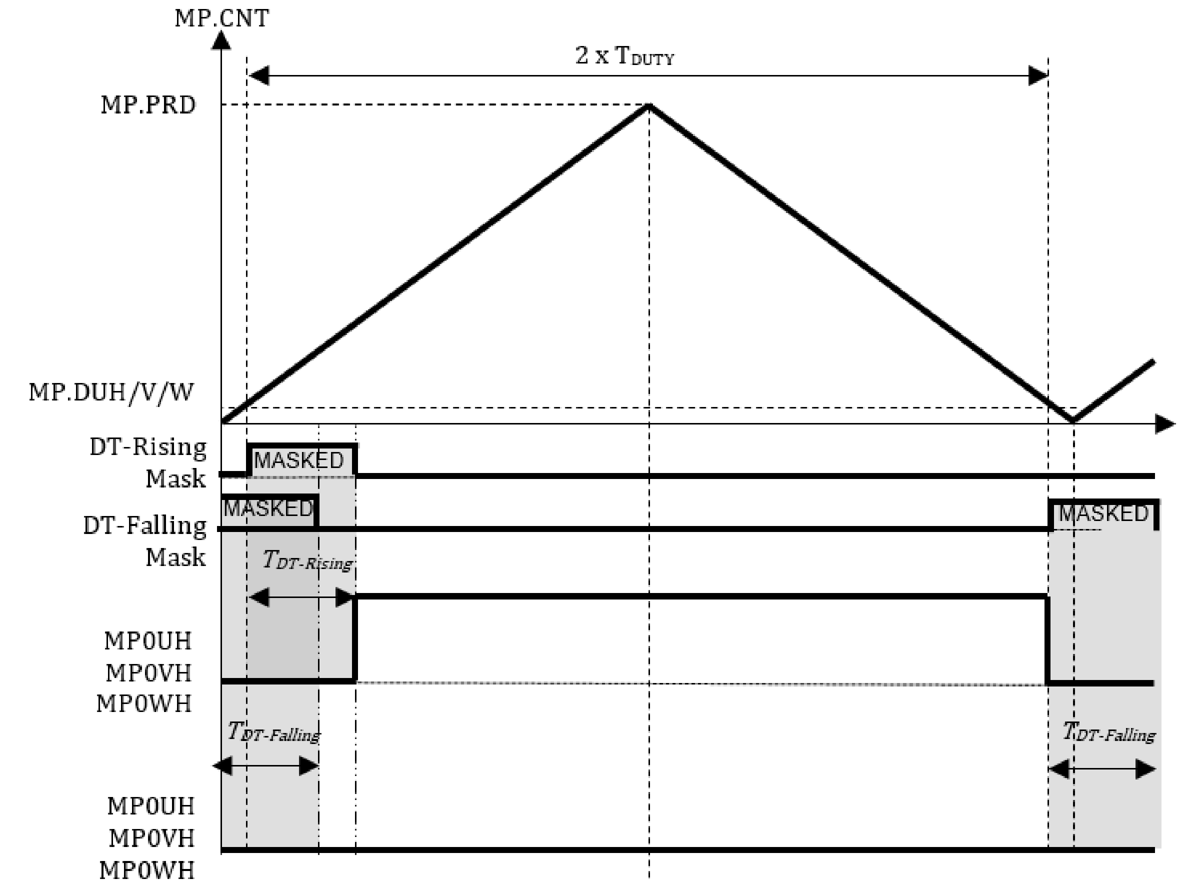


Figure 119. H-side Always On: T_{DUTY} = Period, Dead Time Set Disabled

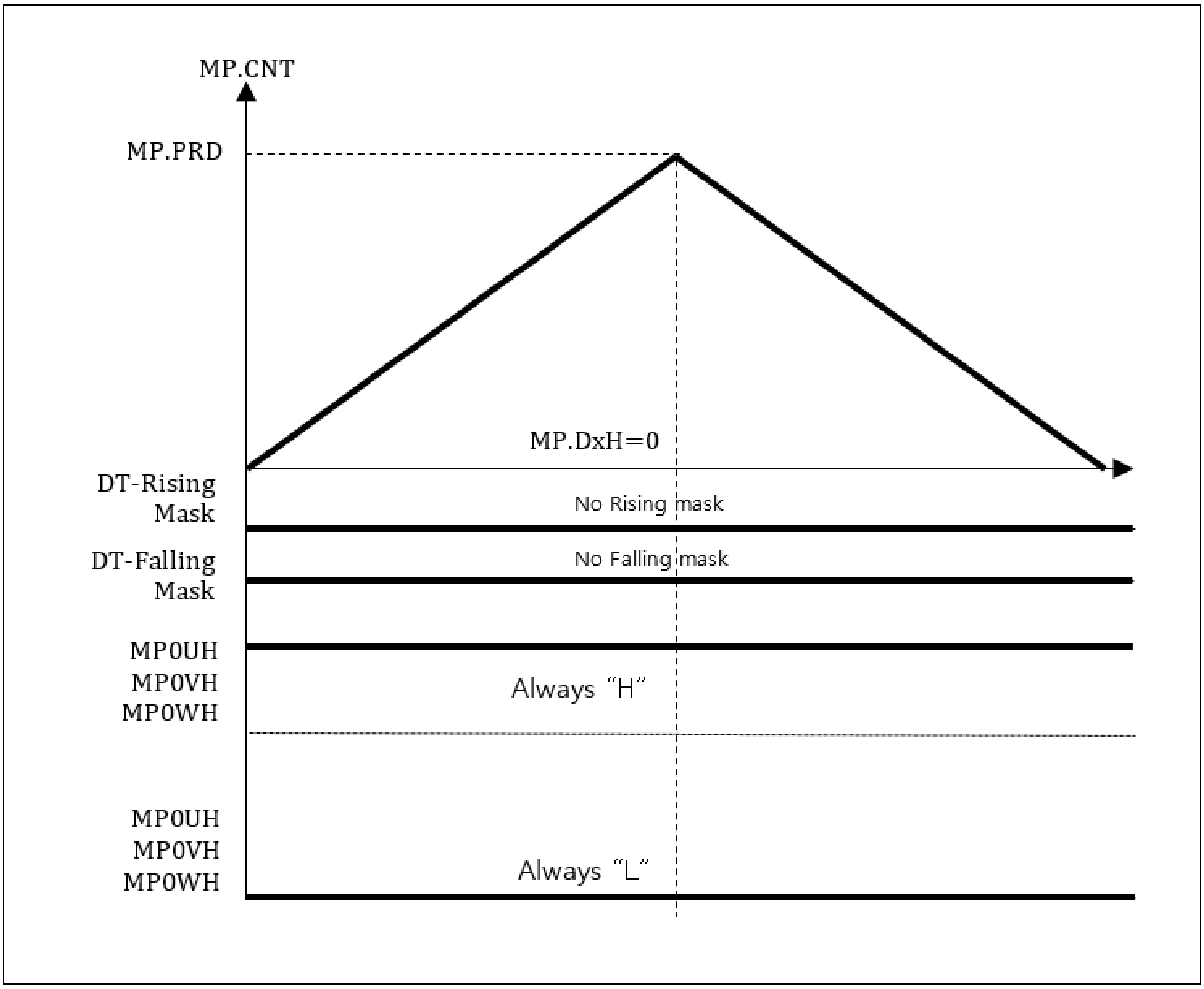
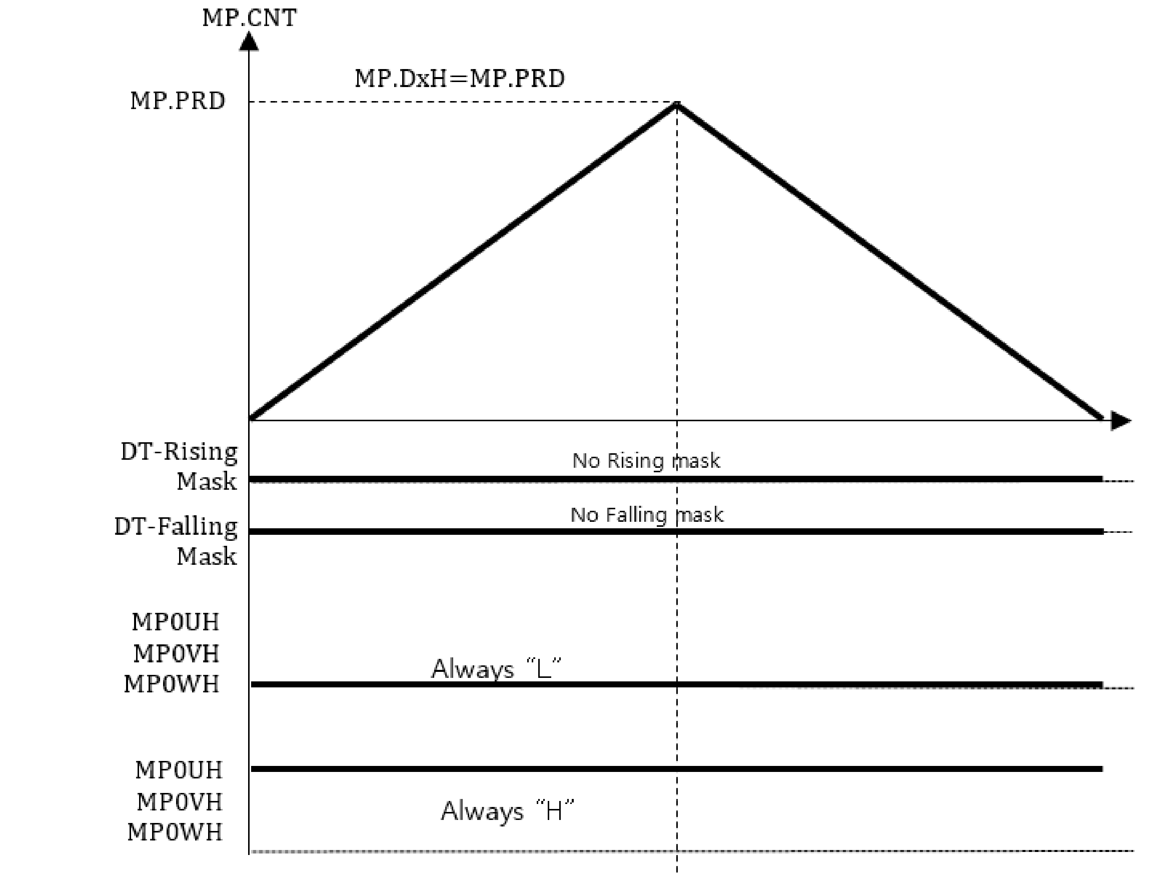


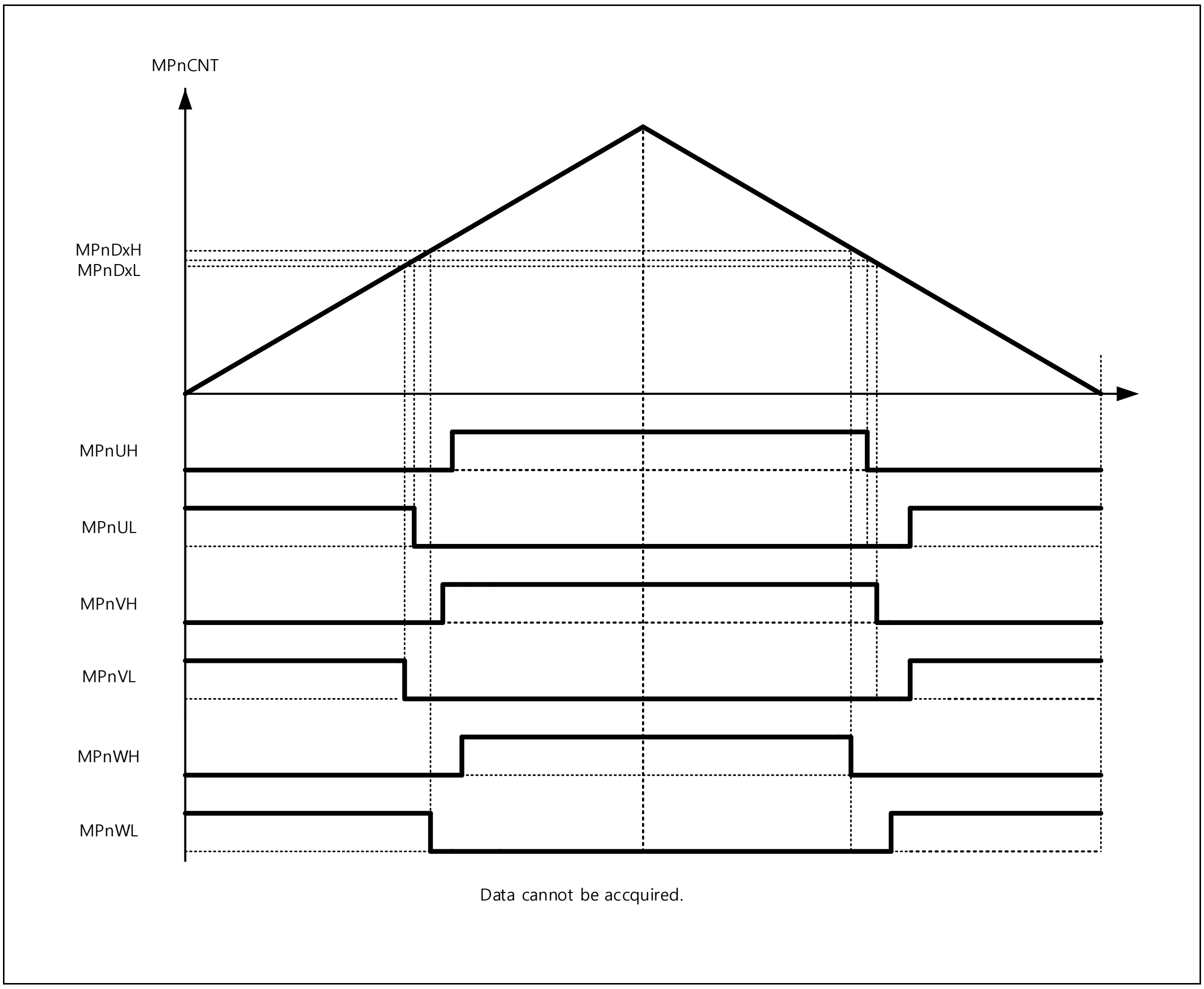
Figure 120. L-side Always On: T_{DUTY} = 0, Dead Time Set Disabled



14.4.11 Symmetric mode vs. asymmetric mode

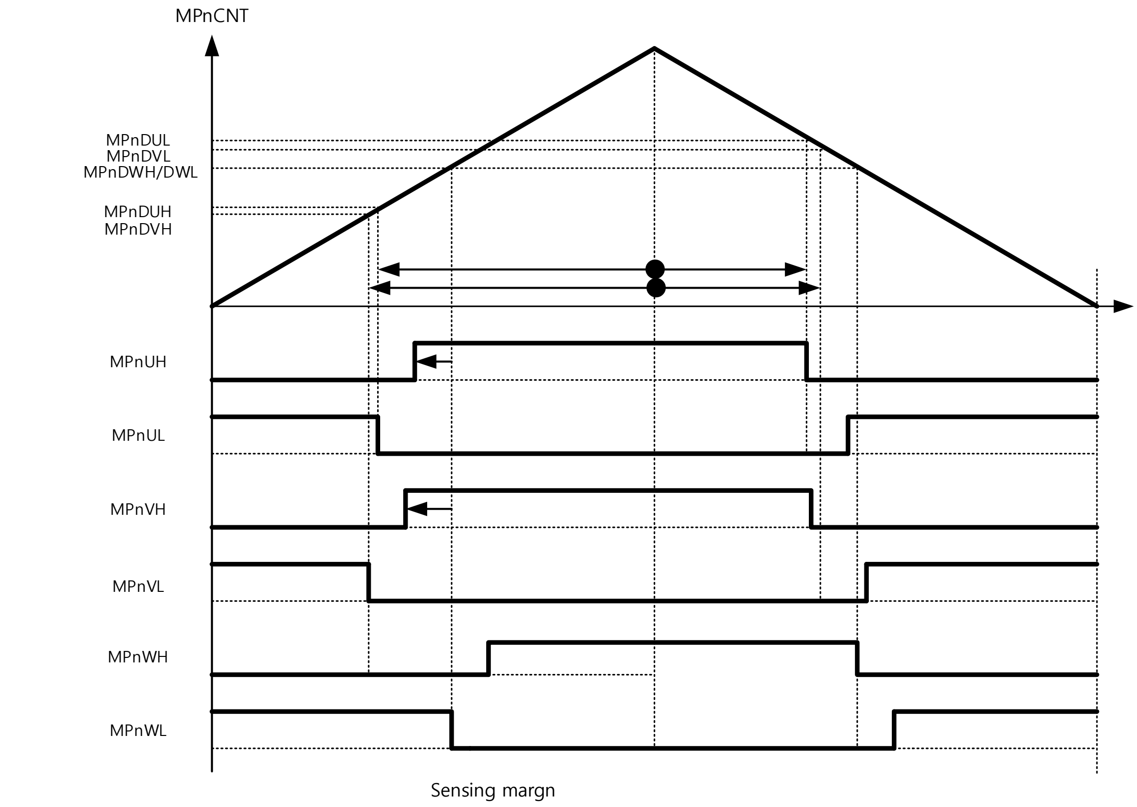
In symmetric mode, the waveform is symmetric with respect to the periodic counter value. Duty comparisons are performed once in the up-count period and once in the down-count period.

Figure 121. Symmetric PWM Timing



In asymmetric mode, the waveform is not symmetrical with respect to the periodic counter value. The duty comparison on the H side is performed in the up-count period. On the L side, it is performed in the down-count period.

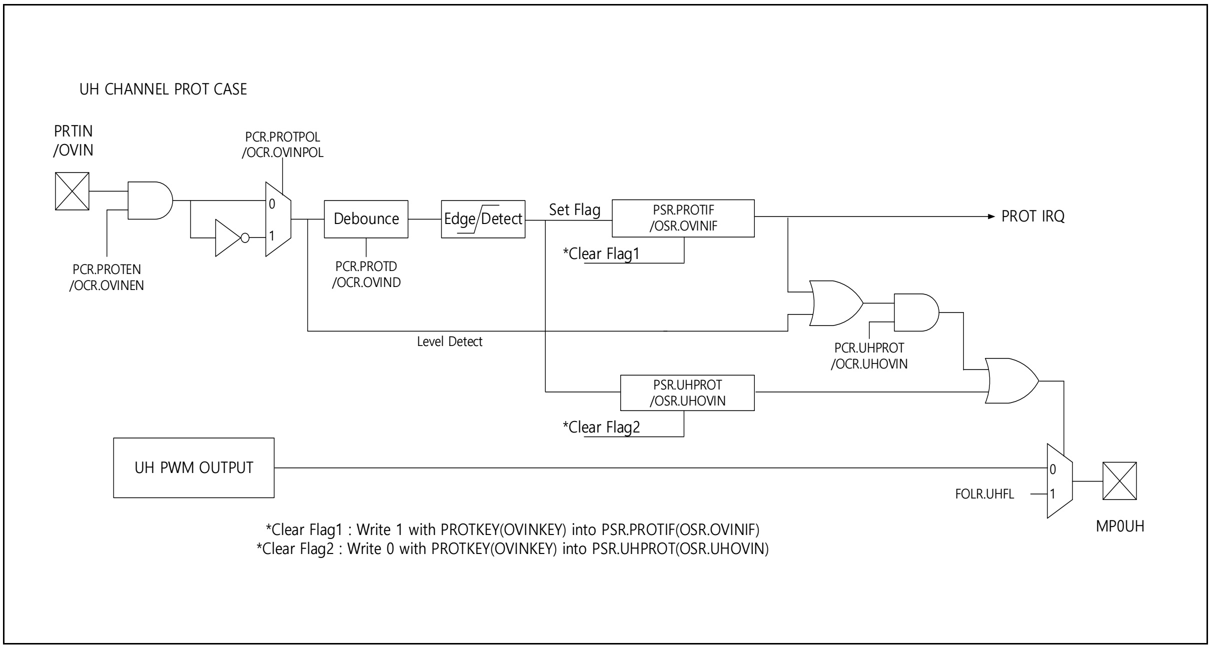
Figure 122. Asymmetric PWM Timing and Limitations in Detection



14.4.12 Functional description of protection and overvoltage

The figure below illustrates how protection and overvoltage are configured in MPWM. The user can forcibly generate a high- or low-level signal from a configured output pin when protection or overvoltage occurs.

Figure 123. Protection and Overvoltage Block Diagram



*Clear Flag1 : Write 1 with PROTKEY(OVINKEY) into PSR.PROTIF(OSR.OVINIF)
*Clear Flag2 : Write 0 with PROTKEY(OVINKEY) into PSR.UHPROT(OSR.UHOVIN)

1. Protection is set in the PCR register, and overvoltage is set in the OCR register.

   In these registers, you can configure various features, including the generation levels of protection and overvoltage, debouncing, and interrupts. Protection and overvoltage can be enabled by setting their respective enable bits.

2. When protection or overvoltage occurs, an output signal is generated in the voltage level set in the FOLR register.

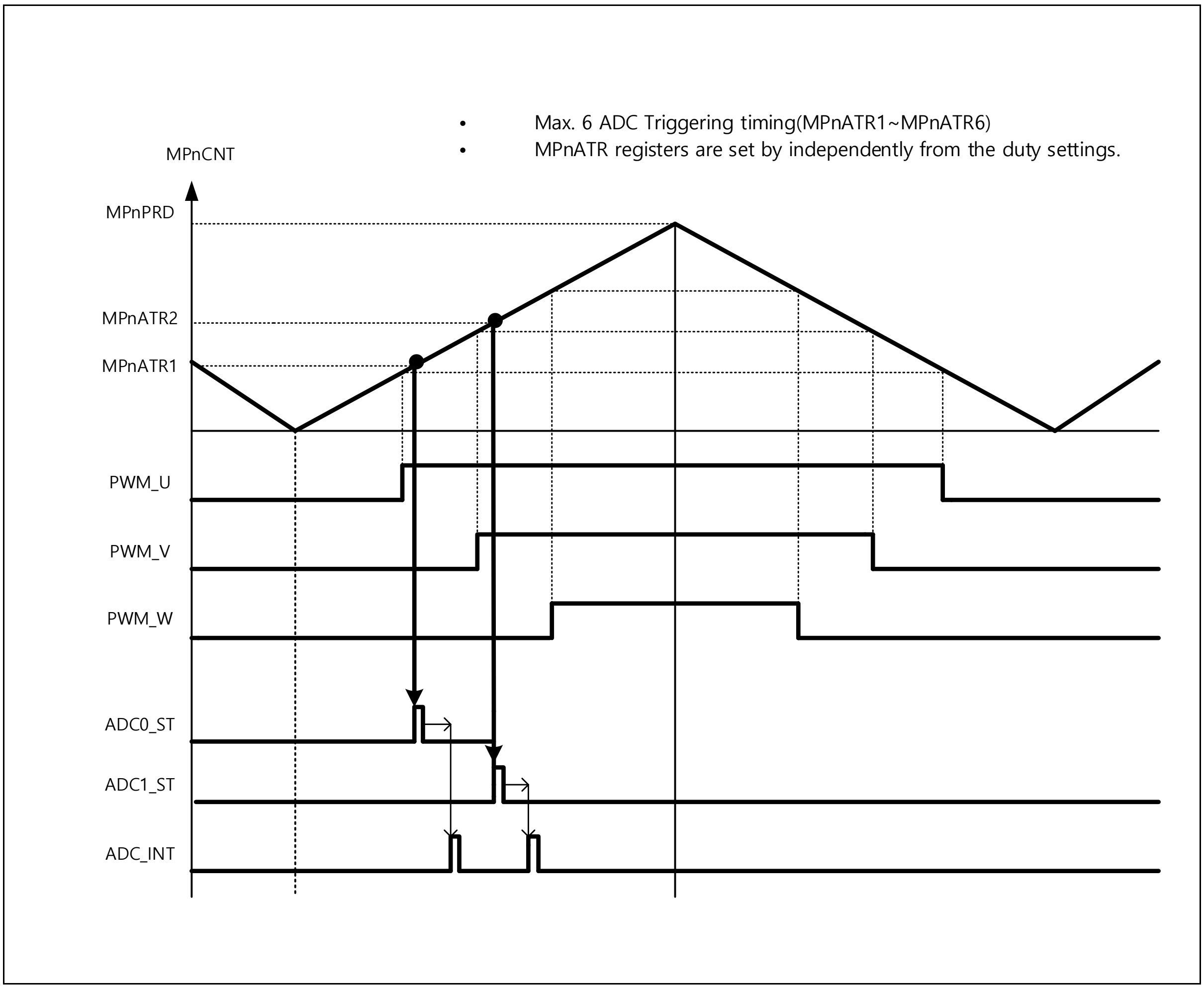
3. In this case, the MPWM protection or overvoltage interrupt occurs, instead of an MPWM interrupt.

4. When such a forced signal is being outputted in the voltage level set in the FOLR register, clearing the corresponding flag in the PSR or OSR register returns the MPWM signal to its original form.

14.4.13 Functional description of ADC triggers

A total of six ADC trigger timing registers are provided. These registers can be used to generate signals that start ADC conversion. The ADC conversion channels can be defined in ADC block registers.

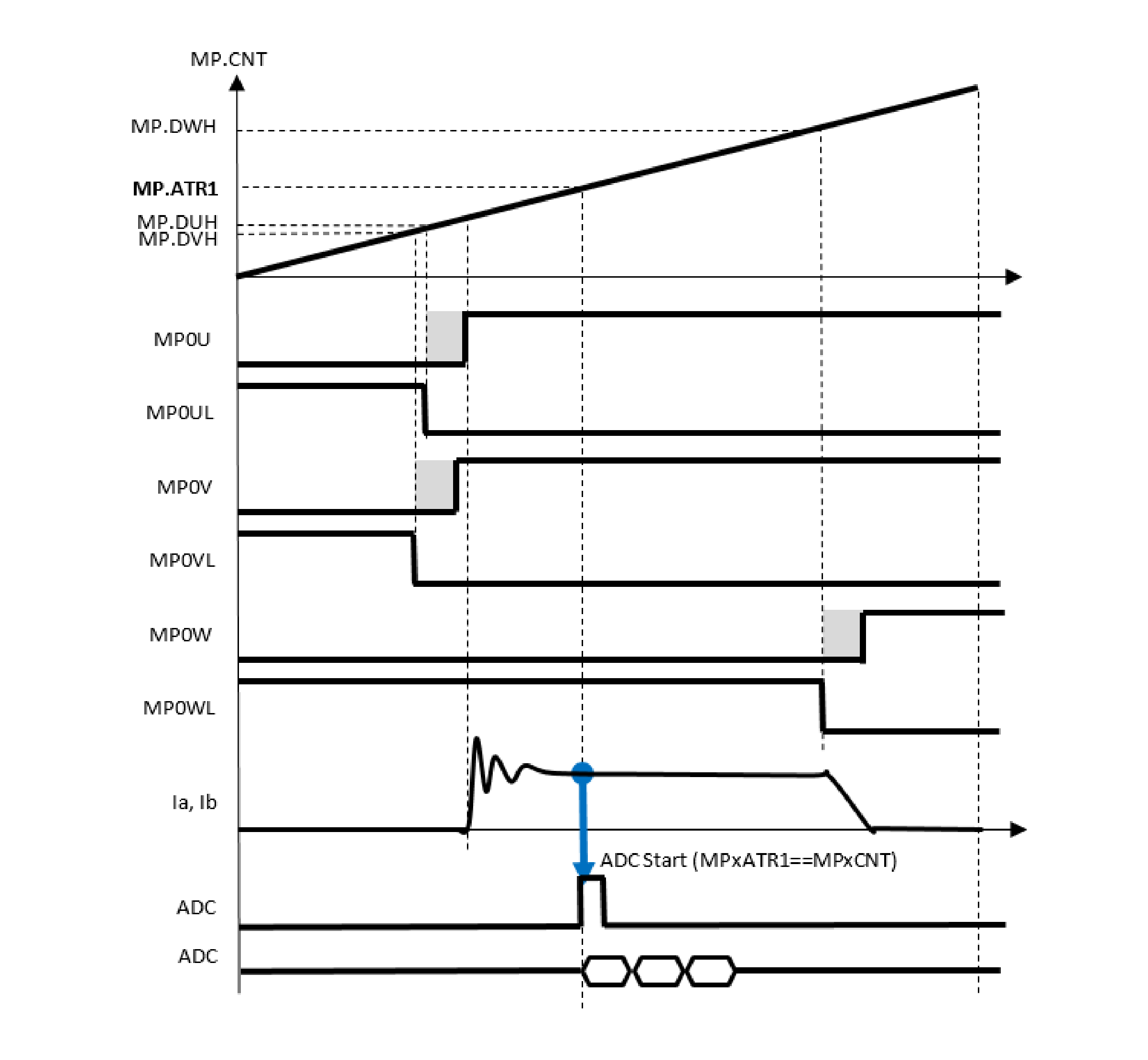
Figure 124. ADC Trigger Timing Diagram



- Max. 6 ADC Triggering timing(MPnATR1~MPnATR6)
- MPnATR registers are set by independently from the duty settings.

The following diagram illustrates the timing of ADC data acquisition.

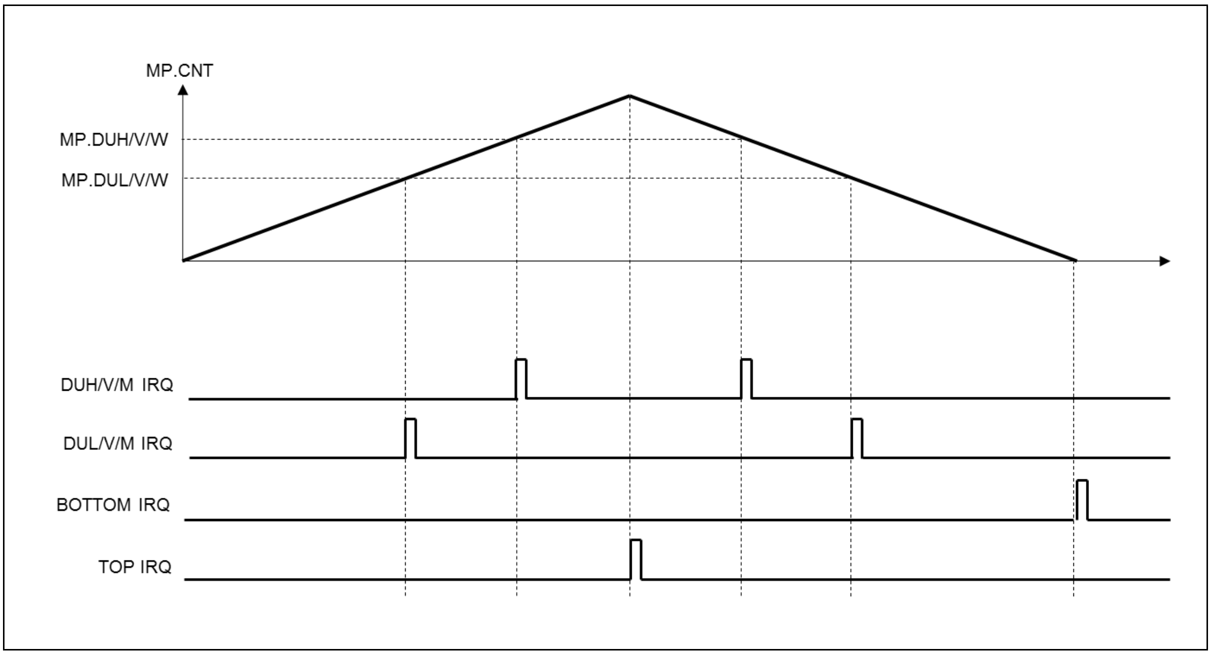
Figure 125. Example Timing of ADC Data Acquisition Triggered by an MPWM Event



14.4.14 Interrupt generation timing

Each timing event can request an interrupt to the CPU.

Figure 126. Interrupt Generation Timing



14.4.15 Setting examples

<Example 1> Initial configuration of MPWM0

```
PORTEN<PORTEN[7:0]> = “0x15”
PORTEN<PORTEN[7:0]> = “0x51”              : Enables PCU register access.
PB_MR1 = “0x00333333”                     : Configures the MPWM0 output pin.
PORTEN<PORTEN[7:0]> = “0”                 : Disables PCU register access.

SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”          : Enables SCU register access.
SCU_MCCR2<MPWM0SEL[10:8]> = “100”
SCU_MCCR2<MPWM0CDIV[7:0]>= “0x01”         : Sets the MPWM0 MCCR clock to MCLK/1.
SCU_PER2<MPWM0[16]> = “1”                 : Enables the MPWM0 peripheral.
SCU_PCER2<MPWM0[16]> = “1”                : Enables the MPWM0 peripheral clock.
SCU_SYSTEN<SYSTEN[7:0]> = “0x00”          : Disables SCU register access.
```

<Example 2> MPWM0 motor mode configuration

```
MPWM0_CR2<PSTART[0]>= “1”                 : Enables the MPWM block.
MPWM0_MR = “0x0091”                       :  Selects  motor  mode,  bottom  match  duty  update,  2-ch  symmetric,  and
                                          up/down counter.
MPWM0_PRD<PERIOD[15:0]>= “0x1000”         : Sets the period value to 0x1000.
MPWM0_DTR = “0x8010”                      : Sets the dead time value to 0x10 and enables the dead time function.

MPWM0_IER = “0x080”                       : Enables the period interrupt.
MPWM0_DUH<PERIOD[15:0]>= “0x0600”         : Sets the UH channel duty to 0x0600.
MPWM0_DUL<DUTY UL[15:0]> = “0x0600”       : Sets the UL channel duty to 0x0600.
MPWM0_DVH<DUTY VH[15:0]> = “0x0700”       : Sets the VH channel duty to 0x0700.
MPWM0_DVL<DUTY VL[15:0]> = “0x0700”       : Sets the VL channel duty to 0x0700.
MPWM0_DWH<DUTY WH[15:0]> = “0x0800”       : Sets the WH channel duty to 0x0800.
MPWM0_DWL<DUTY WL[15:0]> = “0x0800”       : Sets the WL channel duty to 0x0800.
MPWM0_CR1<PWMEN[0]> = “1”                 : Starts MPWM output.
```

<Example 3> MPWM0 protection configuration

```
PORTEN<PORTEN[7:0]> = “0x15”
PORTEN<PORTEN[7:0] = “0x51”               : Enables PCU register access.
PB_MR1 = “0x33333333”                     : Configures the protection pin
PORTEN<PORTEN[7:0] = “0”                  : Disables PCU register access.
MPWM0_FOLR = “0x00”                       : Sets all channels to output low-level signals when protection occurs.
MPWM0_PCR = “0x000080FF”                  : Enables protection (interrupt) for all channels.
while ((MPWM0_PSR&(1<<7) != (1<<7)){      : Checks for the protection (interrupt) flag.
MPWM0_PSR |= ((0xCA<<8)|(0xFF));}         : Clears the flag.
```

<Example 4> Individual mode configuration

```
MPWM0_CR3 = "0x00010101"                  : Enables the U, V, and W channel blocks.
MPWM0_MR = "0xC091"                       : Selects Individual mode, bottom match duty update, 2-ch symmetric, and
                                          up/down counter.

MPWM0_PRDU<PERIOD_U[15:0]> = "0x1000"     : Set the U channel period to 0x1000.
MPWM0_PRDV<PERIOD_V[15:0]> = "0x1200"     : Set the V channel period to 0x1200.
MPWM0_PRDW<PERIOD_W[15:0]> = "0x1400"     : Set the W channel period to 0x1400.
MPWM0_DUH<DUTY UH[15:0]> = "0x0700"       : Sets the UH duty to 0x0700.
MPWM0_DUL<DUTY UL[15:0]> = "0x0900"       : Sets the UL duty to 0x0900
MPWM0_DVH<DUTY VH[15:0]> = "0x0800"       : Sets the VH duty to 0x0800
MPWM0_DVL<DUTY VL[15:0]> = "0x0A00"       : Sets the VL duty to 0x0A00.
MPWM0_DWH<DUTY WH[15:0]> = "0x0A00"       : Sets the WH duty to 0x0A00.
MPWM0_DWL<DUTY WL[15:0]> = 0xA00;         : Sets the WL duty to 0x0A00.
MPWM0_DTRU = "0x801010"                   : Sets the dead time of the U channel.
MPWM0_DTRV = "0x802020                    : Sets the dead time of the V channel
MPWM0_DTRW = "0x803030"                   : Sets the dead time of the W channel.
MPWM0_CR3 = "0x00030303"                  : Starts a simultaneous generation of U, V, and W channel outputs.
```

<Example 5> Capture function configuration in individual mode (only used in the phase U channel)

```
PORTEN<PORTEN[7:0]> = "0x15"
PORTEN<PORTEN[7:0]> = "0x51"              : Enables PCU register access.
PA_MR1 = "0x33300000"                     : Configures the MPWM capture function pin.
PA_MR2 = "0x00033300"                     : Configures the MPWM sub-capture function pin.
PORTEN<PORTEN[7:0]> = "0"                 : Disables PCU register access
MPWM0_CAPCNTU = "0x88000000"              : Initializes and enables the capture counter.
MPWM0_SCAPU = "0x80000000;"               : Initializes the sub-capture counter and sets the sub-capture counter
                                          for rising capture.
```

15 Quadrature Encoder Interface (QEI)

The two-channel Quadrature Encoder Interface (QEI) uses two pulse signals outputted from an encoder. By counting the number of relative phase pulses between these two signals, the encoder's rotational position, direction, and velocity are tracked. Additionally, an index signal is used to reset the position counter.

Each QEI module consists of a decoder logic interpreting the Ph-A and Ph-B signals and up- and down-counters.

The QEI of A33M11x series features the followings:

- Three input pins for two phase signals and index pulse
  - Phases A/B: Input of QEI phases A and B
  - INDEX: Input of QEI index
  - UPDN: Output of phase direction
- 32-bit up-/down-counter counting the number of rotations in each direction
- x2 and x4 count resolution for capture mode
- Position compare register and interrupt
- Index compare register and interrupt
- Velocity capture by a velocity timer
- Signals are selectable
  - Quadrature signals (Ph-A and Ph-B)
  - Clock and direction signals (Clock: Ph-A, direction: Ph-B)

Table 61 introduces pins assigned for QEI interface.

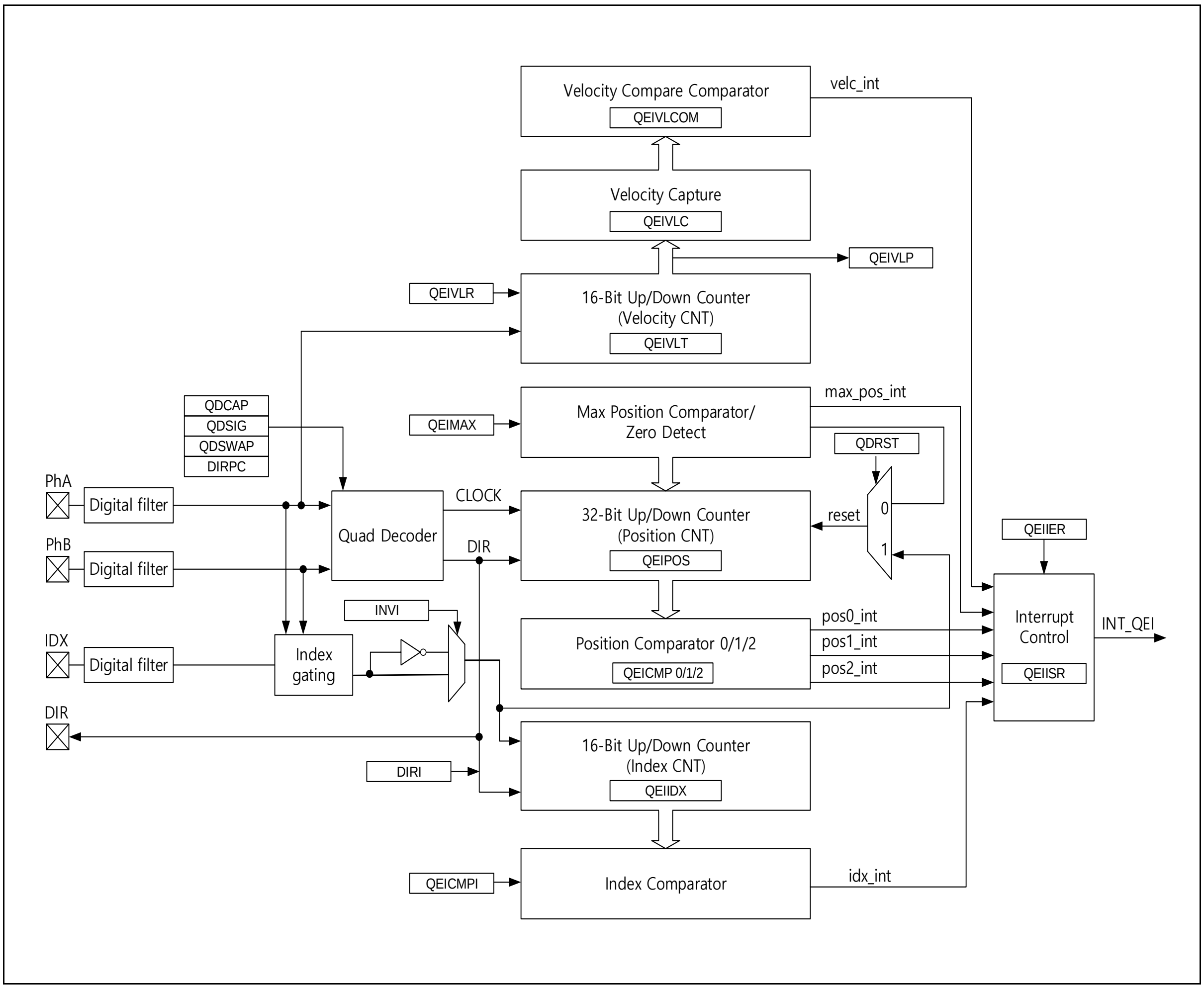
Table 61. Pin Assignment of QEI: External Pins

Pin name	**Type**	**Description**	**Supported packages**	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
QEI0_UPDN	O	QEI0 phase direction output port	O	O
QEI0_A QEI0_B QEI0_IDX	I	QEI0 phase-A, phase-B, and index input ports	O	O
QEI1_UPDN	O	QEI1 phase direction output port	O	X
QEI1_A QEI1_B QEI1_IDX	I	QEI1 phase-A, phase-B, and index input ports	O	X

15.1 QEI block diagram

Figure 127 describes normal mode of MPWM in block diagram.

Figure 127. QEI Block Diagram



15.2 Registers

Base address of QEI is introduced in the followings:

Table 62. Base Address of QEI

Name	Base address
QEI0	0x4000_B400
QEI1	0x4000_B500

Table 63. QEI Register Map

Name	Offset	Type	Description	Reset value	Reference
QEIn_MR	0x0000	RW	QEI n mode register	0x0000_0F00	15.2.1
QEIn_CON	0x0004	RW	QEI n control register	0x0000_0000	15.2.2
QEIn_SR	0x0008	RO	QEI n status register	0x0000_0002	15.2.3
QEIn_POS	0x000C	RW	QEI n position counter register	0x0000_0000	15.2.4
QEIn_MAX	0x0010	RW	QEI n maximum position register	0xFFFF_FFFF	15.2.5
QEIn_CMP0	0x0014	RW	QEI n position compare 0 register	0xFFFF_FFFF	15.2.6
QEIn_CMP1	0x0018	RW	QEI n position compare 1 register	0xFFFF_FFFF	15.2.7
QEIn_CMP2	0x001C	RW	QEI n position compare 2 register	0xFFFF_FFFF	15.2.8
QEIn_IDX	0x0020	RW	QEI n index counter register	0x0000_0000	15.2.9
QEIn_CMPI	0x0024	RW	QEI n index compare register	0x0000_FFFF	15.2.10
QEIn_VLR	0x0030	RW	QEI n velocity reload register	0x0000_FFFF	15.2.11
QEIn_VLT	0x0034	RW	QEI n velocity timer register	0x0000_FFFF	15.2.12
QEIn_VLP	0x0038	RW	QEI n velocity pulse counter register	0x0000_0000	15.2.13
QEIn_VLC	0x003C	RW	QEI n velocity capture register	0x0000_FFFF	15.2.14
QEIn_VLCOM	0x0040	RW	QEI n velocity compare register	0x0000_0000	15.2.15
QEIn_IER	0x0050	RW	QEI n interrupt enable register	0x0000_0000	15.2.16
QEIn_ISR	0x0054	RO	QEI n interrupt status register	0x0000_0000	15.2.17
QEIn_ISCR	0x0058	WO	QEI n interrupt status clear register	0x0000_0000	15.2.18

NOTE: n = 0 and 1

15.2.1 QEIn_MR: QEI n mode register

QEIn_MR includes bits that determine the operations of the QEI module.

QEI0_MR=0x4000_B400, QEI1_MR=0x4000_B5004

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	INXGATE	QDVEL	DIRI	DIRPC	QDRST	QDCAP	QDSIG	QDSWAP	QDMOD
-	1111	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	RW	RW	RW

11 8	INXGATE	Index gating configuration	
		1111	Passes the index.
		1000	Passes the index when Ph-A = 0 and Ph-B = 0.
		0100	Passes the index when Ph-A = 0 and Ph-B = 1.
		0010	Passes the index when Ph-A = 1 and Ph-B = 1.
		0001	Passes the index when Ph-A = 1 and Ph-B = 0.
7	QDVEL	Whether to enable or disable the velocity counter	
		0	Disables.
		1	Enables.
6	DIRI	Index counter direction control	
		0	DIRECTION status does not affect the counter.
		1	DIRECTION status changes the count direction.
5	DIRPC	Position counter direction control	
		0	DIRECTION status does not affect the counter.
		1	DIRECTION status changes the count direction.
4	QDRST	Position counter reset mode configuration	
		0	Reset is triggered by the maximum position reset.
		1	Reset is triggered by an index pulse.
3	QDCAP	Capture mode configuration (X2 or X4)	
		0	Only PhA edge is counted
		1	PhA and PhB edges are counted
2	QDSIG	Signal mode configuration	
		0	Quadrature phase signals (Ph-A and Ph-B)
		1	Clock and direction signals (clock: Ph-A, direction: Ph-B)
1	QDSWAP	QEI input signal SWAP configuration	
		0	No Swap
		1	Swap Ph-A and Ph-B
0	QDMOD	Whether to enable or disable the QEI module	
		0	Disables.
		1	Enables.

15.2.2 QEIn_CON: QEI n control register

QEIn_CON includes bits that control the QEI module’s position and velocity counters.

QEI0_CON=0x4000_B404, QEI1_CON=0x4000_B504

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">Reserved</td><td>INVI</td><td colspan="5">Reserved</td><td>RESV</td><td>RESI</td><td>RESP</td></tr>
<tr><td colspan="23">-</td><td>0</td><td colspan="5">-</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="23">-</td><td>RW</td><td colspan="5">-</td><td>RW</td><td>RW</td><td>RW</td></tr>
</table>

8	INVI	Index pulse inversion	
		0	None
		1	Inverts the index pulse.
2	RESV	Velocity counter initialization	
		0	None
		1	Initializes the velocity counter.
1	RESI	Index counter initialization	
		0	None
		1	Initializes the index counter.
0	RESP	Position counter initialization	
		0	None
		1	Initializes the position counter.

NOTES:

1. If the RESV, RESI, and RESP bits are set to 1, all the counters are reset to 0 and reloaded.
2. Exceptionally when the RESV bit is set to '1', the QEIn_VLC register value must be initialized to 0xFFFF.

15.2.3 QEIn_SR: QEI n status register

QEIn_SR displays the status of the QEI module.

QEI0_SR=0x4000_B408, QEI1_SR=0x4000_B508

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">Reserved</td><td>DIRECTION</td><td>ERROR</td></tr>
<tr><td colspan="30">-</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">-</td><td>RO</td><td>RO</td></tr>
</table>

1	DIRECTION	Rotation direction status	
		0	The motor rotates in reverse direction.
		1	The motor rotates in forward direction.
0	ERROR	An error was detected in the gray code sequence (both signals are changed at the same time)	

15.2.4 QEIn_POS: QEI n position counter register

QEIn_POS is a position counter register of the QEI module. This contains the current encoder position value. The counter increases or decreases depending on rotational direction.

QEI0_POS=0x4000_B40C, QEI1_POS=0x4000_B50C

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
QEIPOS																															
0x0000_0000																															
RW																															

31 0	QEIPOS	Current position counter value

15.2.5 QEIn_MAX: QEI n maximum position register

QEIn_MAX is a position counter maximum register of the QEI module. During forward rotation, once this value is exceeded by the QEIn_POS register value, QEIn_POS is reset to zero. During reverse rotation, the QEIn_POS register value is reset to this register value when QEIn_POS decreases from zero.

QEI0_MAX=0x4000_B410, QEI1_MAX=0x4000_B510

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
QEIMAX																															
0xFFFF_FFFF																															
RW																															

31 0	QEIMAX	The maximum value of the position counter

15.2.6 QEIn_CMP0: QEI n position compare 0 register

QEIn_CMP0 is a position counter compare 0 register of the QEI module. This register value is compared with the QEIn_POS register’s current value. When the two registers have the same value, an interrupt occurs.

QEI0_CMP0=0x4000_B414, QEI1_CMP0=0x4000_B514

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
QEICMP0			
0xFFFF_FFFF			
RW			

31 0	QEICMP0	Position compare value 0

15.2.7 QEIn_CMP1: QEI n position compare 1 register

QEIn_CMP1 is a position counter compare 1 register of the QEI module. This register value is compared with the QEIn_POS register’s current value. When the two registers have the same value, an interrupt occurs.

QEI0_CMP1=0x4000_B418, QEI1_CMP1=0x4000_B518

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
QEICMP1			
0xFFFF_FFFF			
RW			

31 0	QEICMP1	Position compare value 1

15.2.8 QEIn_CMP2: QEI n position compare 2 register

QEIn_CMP2 is a position counter compare 2 register of the QEI module. This register value is compared with the QEIn_POS register's current value. When the two registers have the same value, an interrupt occurs.

QEI0_CMP2=0x4000_B41C, QEI1_CMP2=0x4000_B51C

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
QEICMP2			
0xFFFF_FFFF			
RW			

31 0	QEICMP2	Position compare value 2

15.2.9 QEIn_IDX: QEI n index counter register

QEIn_IDX is an index counter register of the QEI module. The encoder counter increases or decreases depending on rotational direction.

QEI0_IDX=0x4000_B420, QEI1_IDX=0x4000_B520

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		QEIIDX	
-		0x0000	
-		RW	

15 0	QEIIDX	Current index counter value

15.2.10 QEIn_CMPI: QEI n index compare register

QEIn_CMPI is an index counter compare register of the QEI module. This register value is compared with the QEIn_IDX value. When the two registers have the same value, an interrupt occurs.

QEI0_CMPI=0x4000_B424, QEI1_CMPI=0x4000_B524

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		QEICMPI	
-		0xFFFF	
-		RW	

15 0	QEICMPI	Index counter compare value

15.2.11 QEIn_VLR: QEI n velocity reload register

QEIn_VLR is a velocity reload register of the QEI module. This register’s set value is reloaded to the velocity timer register (QEin_VLT) when the QEin_VLT value becomes zero.

QEI0_VLR=0x4000_B430, QEI1_VLR=0x4000_B530

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		QEIVLR	
-		0xFFFF	
-		RW	

15 0	QEIVLR	Velocity timer reload value

15.2.12 QEIn_VLT: QEI n velocity timer register

The QEIn_VLT is a velocity timer register of the QEI module. Once the timer reaches zero, the velocity pulse register (QEIn_VLP) value is stored in the velocity capture register (QEIn_VLC). And the velocity reload register (QEIn_VLR) value is reloaded.

QEI0_VLT=0x4000_B434, QEI1_VLT=0x4000_B534

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		QEIVLT	
-		0xFFFF	
-		RW	

15 0	QEIVLT	Velocity timer’s timer value

15.2.13 QEIn_VLP: QEI n velocity pulse counter register

QEIn_VLP is a velocity pulse counter register of the QEI module. This register includes the number of velocity pulses counted during the current timer cycle. This register value is captured in the velocity capture register (QEIn_VLC) when the velocity timer register (QEIn_VLT) value becomes zero. After the capturing, this register value becomes zero.

QEI0_VLP=0x4000_B438, QEI1_VLP=0x4000_B538

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		QEIVLP	
-		0x0000	
-		RW	

15 0	QEIVLP	Current velocity pulse counter value

15.2.14 QEIn_VLC: QEI n velocity capture register

QEIn_VLC is a velocity capture register of the QEI module. This register indicates the number of pulses counted during the velocity timer cycle.

QEI0_VLC=0x4000_B43C, QEI1_VLC=0x4000_B53C

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		QEIVLC	
-		0xFFFF	
-		RW	

15 0	QEIVLC	Current velocity capture value

15.2.15 QEIn_VLCOM: QEI n velocity compare register

QEIn_VLCOM is a velocity compare register of the QEI module. This register value is compared with the velocity captured in the velocity capture register (QEIn_VLC). When the captured velocity is lower than this compare value, an interrupt occurs if the interrupt is set enabled.

QEI0_VLCOM=0x4000_B440, QEI1_VLCOM=0x4000_B540

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		QEIVLCOM	
-		0x0000	
-		RW	

15 0	QEIVLCOM	Velocity compare value

15.2.16 QEIn_IER: QEI n interrupt enable register

QEIn_IER include bits that determine the enablement of each interrupt of the QEI module.

QEI0_IER=0x4000_B450, QEI1_IER=0x4000_B550

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	VELCEN	VELTEN	IDXEN	MAXEN	POS2EN	POS1EN	POS0EN	ENCLKEN	ERREN	DIREN	INXEN
-	0	0	0	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

10	VELCEN	Whether to enable or disable the interrupt triggered when a capture value smaller than the velocity compare value is entered	
		0	Disables.
		1	Enables.
9	VELTEN	Whether to enable or disable the interrupt triggered when the velocity timer value becomes zero	
		0	Disables.
		1	Enables.
8	IDXEN	Whether to enable or disable the interrupt triggered when the index counter value equals the compare value	
		0	Disables.
		1	Enables.
7	MAXEN	Whether to enable or disable the interrupt triggered when the position counter value equals the set maximum value	
		0	Disables.
		1	Enables.
6 5 4	POSnEN (n=0–2)	Whether to enable or disable the interrupt triggered when the position n compare register value equals the current counter value	
		0	Disables.
		1	Enables.
3	ENCLKEN	Whether to enable or disable the decoder clock pulse generation flag	
		0	Disables.
		1	Enables.
2	ERREN	Whether to enable or disable the decoder phase error flag	
		0	Disables.
		1	Enables.
1	DIREN	Whether to enable or disable the direction change flag	
		0	Disables.
		1	Enables.
0	INXEN	Whether to enable or disable the index pulse generation flag	
		0	Disables.
		1	Enables.

15.2.17 QEIn_ISR: QEI n interrupt status register

QEIn_ISR includes bits that display the status of the QEI module.

QEI0_ISR=0x4000_B454, QEI1_ISR=0x4000_B554

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	VELC	VELT	IDX	MAX	POS2	POS1	POS0	ENCLK	ERR	DIR	INX
-	0	0	0	0	0	0	0	0	0	0	0
-	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

10	VELC	Flag indicating that a capture value smaller than the velocity compare value has been entered	
		0	Not flagged.
		1	Flagged.
9	VELT	Flag indicating that the velocity timer value has become zero	
		0	Not flagged.
		1	Flagged.
8	IDX	Flag indicating that the index counter value has equaled the compare value	
		0	Not flagged.
		1	Flagged.
7	MAX	Flag indicating that the position counter value has equaled the set maximum value	
		0	Not flagged.
		1	Flagged.
6 5 4	POSn (n=0–2)	Flag indicating that the position n compare register value has equaled the current counter value	
		0	Not flagged.
		1	Flagged.
3	ENCLK	Decoder clock pulse generation flag	
		0	Not flagged.
		1	Flagged.
2	ERR	Decoder phase error flag	
		0	Not flagged.
		1	Flagged.
1	DIR	Direction change flag	
		0	Not flagged.
		1	Flagged.
0	INX	Index pulse generation flag	
		0	Not flagged.
		1	Flagged.

15.2.18 QEIn_ISCR: QEI n interrupt status clear register

QEIn_ISCR is used to clear interrupt status flags of the QEI module. A flagged bit is cleared by being written to a 1. If you try to read the register, it will be read as 0x00.

QEI0_ISCR=0x4000_B458, QEI1_ISCR=0x4000_B558

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">Reserved</td><td>VELC</td><td>VELT</td><td>IDX</td><td>MAX</td><td>POS2</td><td>POS1</td><td>POS0</td><td>ENCLK</td><td>ERR</td><td>DIR</td><td>INX</td></tr>
<tr><td colspan="21">-</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="21">-</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td><td>WO</td></tr>
</table>

10	VELC	Flag indicating that a capture value smaller than the velocity compare value has been entered	
		0	Causes no changes.
		1	Clears the flag.
9	VELT	Flag indicating that the velocity timer value has become zero	
		0	Causes no changes.
		1	Clears the flag.
8	IDX	Flag indicating that the index counter value has equaled the compare value	
		0	Causes no changes.
		1	Clears the flag.
7	MAX	Flag indicating that the position counter value has equaled the set maximum value	
		0	Causes no changes.
		1	Clears the flag.
6 5 4	POSn (n=0–2)	Flag indicating that the position n compare register value has equaled the current counter value	
		0	Causes no changes.
		1	Clears the flag.
3	ENCLK	Decoder clock pulse generation flag	
		0	Causes no changes.
		1	Clears the flag.
2	ERR	Decoder phase error flag	
		0	Causes no changes.
		1	Clears the flag.
1	DIR	Direction change flag	
		0	Causes no changes.
		1	Clears the flag.
0	INX	Index pulse generation flag	
		0	Causes no changes.
		1	Clears the flag.

15.3 Functional description

15.3.1 Quadrature input signals

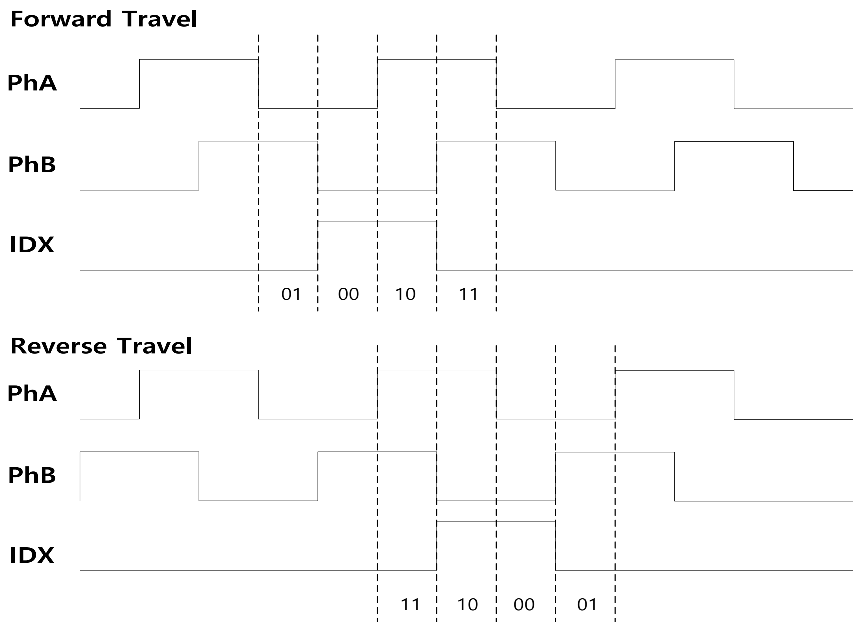
Each QEI module uses a number of signals for two operating modes: quadrature phase mode and clock/direction mode. These modes can be selected by setting QEIn_MR's QDSIG bit.

In quadrature phase mode, the encoder generates two clocks (Ph-A and Ph-B) whose phase difference is 90°; the two clock signals' edge sequence is used to determine the rotational direction.

In clock/direction mode, a clock signal (Ph-A) and a rotational direction signal (Ph-B) are used.

The position counter increments when the Ph-A edge signal is received by the QEI module earlier than the Ph-B edge signal. And, the position counter decrements when the Ph-B edge signal is received earlier than the Ph-A signal.

Figure 128. State Changes in the Encoder



The formula below calculates the input encoder frequency:

$$\text{Input Encoder Frequency} = \left(\frac{\text{RPM}}{60}\right) \times (Slot_{number} \times 4)$$

(Input encoder frequency < max capture frequency)

The formula below calculates the maximum capture frequency (The input encoder frequency must not exceed this maximum capture frequency):

$$\text{Max Capture Frequency} = \left(\frac{\text{PCLK}}{4 \times 2}\right)$$

<Example>

In the denominator, “4” represents the number of debounces and “2” represents the number of captures, one at the rising edge and the other at the falling edge.

Table 64. Encoder States

Phase A	Phase B	State
0	1	1
0	0	2
1	0	3
1	1	4

Table 65. Encoder State Changes

From State	To State	Direction
1	2	Forward
2	3	
3	4	
4	1	
4	3	Reverse
3	2	
2	1	
1	4	

Table 66. Encoder Direction Changes

DIRI bit	INVI bit	Direction
0	0	Forward
1	0	Reverse
0	1	Reverse
1	1	forward

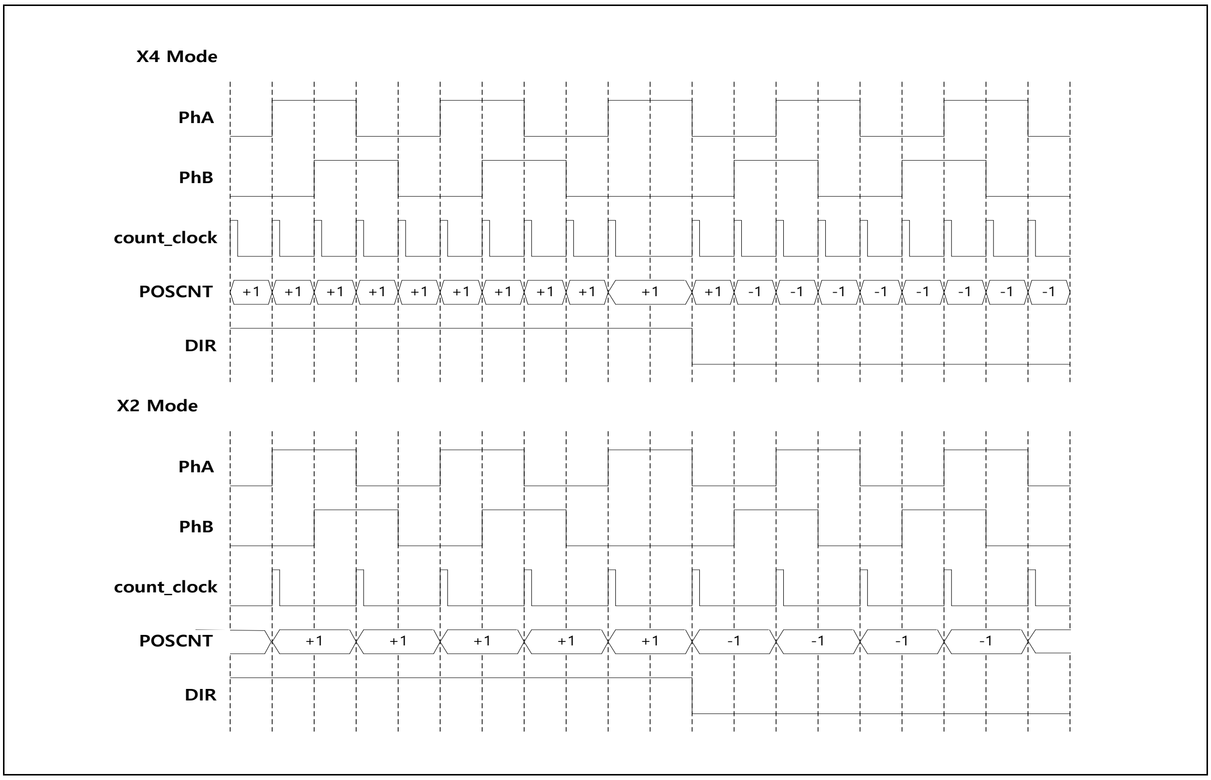
15.3.2 Position capture

In position capture mode, you can configure the position counter to be updated at every edge of the Ph-A signal or at every edge of both the Ph-A and Ph-B signals.

Updating for both phases (Ph-A and Ph-B) narrows the counter range but makes the position measurement resolution finer.

Capture mode edge counting can be configured with the QEIn_MR.QDCAP bit.

Figure 129. Capture Mode Edge Configuration: X2 and X4 Modes

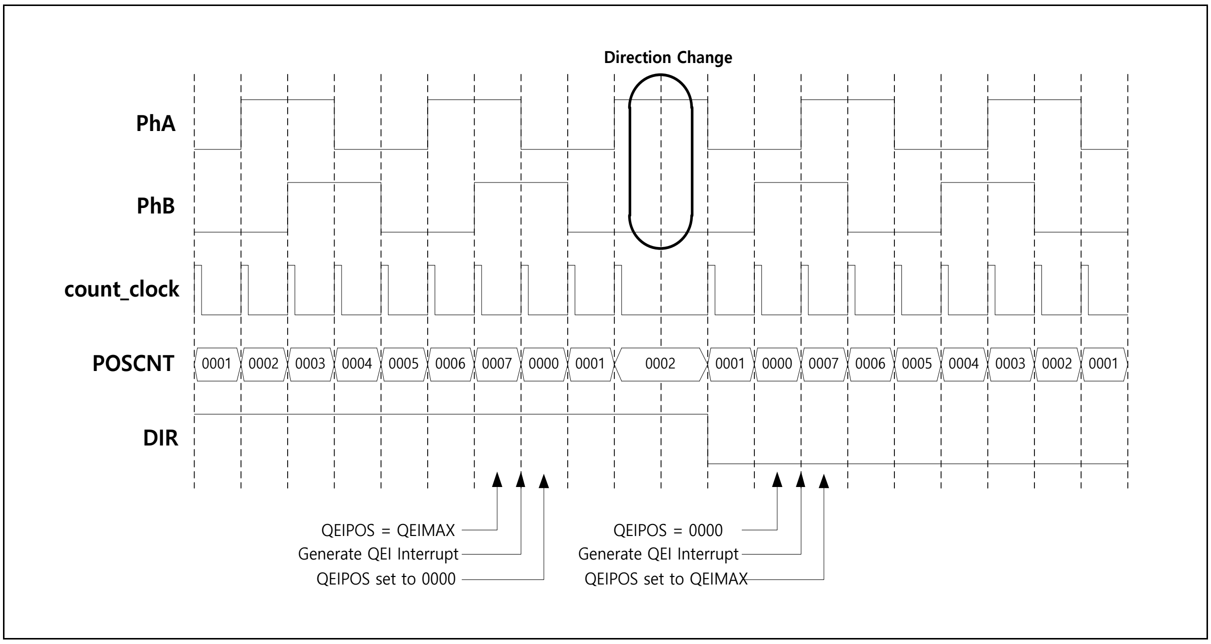


1. QEIMAX-Triggering Position Counter Reset

During the encoder's rotation in forward direction (Ph-A precedes Ph-B), the QEIn_POS value counts up by 1 each time. Once the QEIn_POS value reaches the QEIn_MAX register's set value, QEIn_POS is reset to zero at the next edge.

If the encoder shifts its rotational direction from forward to reverse, the QEIn_POS value starts counting down by 1. Once the QEIn_POS value reaches zero, the QEI_MAX register's set value is loaded to QEIn_POS.

Figure 130. Position Counter Reset: QEI_MAX Register Used



2. Index-Triggering Position Counter Reset

   Setting the QEIn_MR register's QDRST to 1 has the position counter initialized at the occurrence of an index pulse. In this mode, the position counter resetting mechanism works as follows:

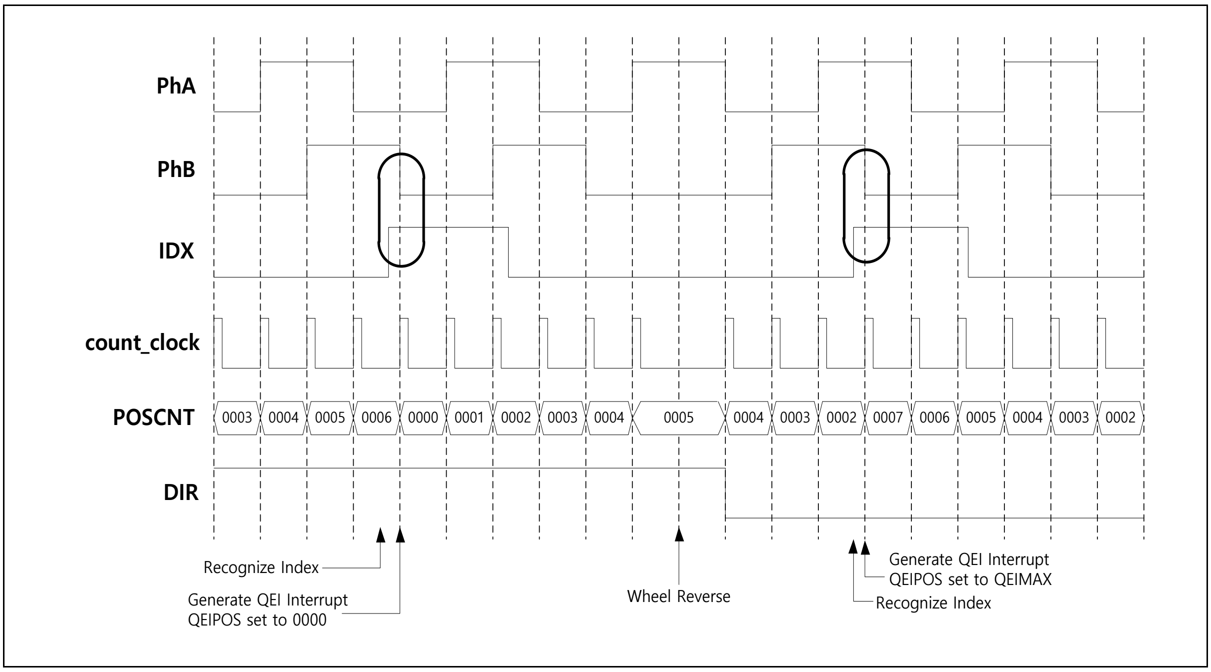
   A. The position counter is initialized each time an index pulse occurs at the index pin.

   B. The counter's initialized value varies with the encoder's rotational direction.

   C. If an index pulse occurs during forward rotation, the QEIn_POS value is initialized to zero and starts counting up by 1.

   D. An index pulse during reverse rotation triggers QEIn_MAX's set value to be loaded, from which QEIn_POS starts counting down by 1.

Figure 131. Position Counter Reset: Index Pulse Used



15.3.3 Velocity capture

The velocity capture function uses a programmable timer and a capture register. The timer is used to count the number of phase edges detected in a given period (The same function as position counter).

The capturing function allows the velocity pulse counter register (QEIn_VLP) value to be transferred to the capture register (QEIn_VLC) when the velocity counter register (QEIn_VLT) value becomes zero. On the completion of this value transfer, the velocity reload register (QEIn_VLR) value is loaded. And finally, the velocity interrupt bit (QEIn_ISR's VELT bit) becomes flagged. The number of edge counts during the period set for the timer or capturing is directly proportional to the speed of the encoder.

The following formula converts the velocity counter value to an RPM value:

$$\mathrm{RPM} = (\mathrm{PCLK} \times \mathrm{Speed} \times 60)/\,(\mathrm{Load} \times PPR \times \mathrm{Edges})$$

NOTES:

1. PCLK is the QEI clock.
2. PPR represents the number of pulses per rotation of the physical encoder.
3. The number of edges is determined by the QDCAP setting (X2 or X4) of QEIMR.

15.3.4 Velocity compare value

The velocity timer uses the velocity compare value register QEIn_VLCOM. After a velocity capture event, the value of the velocity capture register QEIn_VLC is compared to the value of the velocity compare register QEIn_VLCOM. If the captured velocity value is less than the compare register value, the velocity compare interrupt occurs (if the interrupt has been set enabled). This function can be used to check if the motor doesn't move or runs too slowly.

15.3.5 Setting examples

<Example 1> QEI0 position counter configuration

```
SCU_SYSTEN<STSTEN[7:0]> = “0x57”
SCU_SYSTEN<STSTEN[7:0]> = “0x75”                : Unlocks the SCU registers.
SCU_PER1<QEI0[28]> = “1”                        : Enables the QEI0 peripheral.
SCU_PCER1<QEI0[28]> = “1”                       : Enables the QEI0 peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                 : Enables PORTEN (enter 0x15 and then 0x51).

PA_MR2<P12MUX[18:16]> = “011”                   : Sets Port A pin P12 as QEI0_UPDN.
PA_MR2<P13MUX[22:20]> = “011”                   : Sets Port A pin P13 as QEI0_A.
PA_MR2<P14MUX[26:24]> = “011”                   : Sets Port A pin P14 as QEI0_B.
PA_MR2<P15MUX[30:28]> = “011”                   : Sets Port A pin P15 as QEI0_IDX.
PA_CR<P12[25:24]> = “00”                        : Sets Port A pin P12 as an output pin.
PA_CR<P13[27:26]> = “11”                        : Sets Port A pin P13 as an input pin.
PA_CR<P14[29:28]> = “11”                        : Sets Port A pin P14 as an input pin.
PA_CR<P15[31:30]> = “11”                        : Sets Port A pin P15 as an input pin.
PA_PRCR<PUE12[25:24]> = “00”                    : Disables pull-up/pull-down at Port A pin P12.
PA_PRCR<PUE13[27:26]> = “00”                    : Disables pull-up/pull-down at Port A pin P13.
PA_PRCR<PUE14[29:28]> = “00”                    : Disables pull-up/pull-down at Port A pin P14.
PA_PRCR<PUE15[31:30]> = “00”                    : Disables pull-up/pull-down at Port A pin P15.

NVICIP[31]<PRI_31[31:24]> = “01110000”          : Sets the NVIC QEI0 interrupt’s priority level.
NVICISER[0]<SETPEND[31:0]>                      : Enables the NVIC QEI0 interrupt.
= “10000000_00000000_00000000_00000000”

QEI0_MR<INXGATE[11:8]> = “0100”                 : Sets the index gates (Ph-A = 0 and Ph-B = 1).
QEI0_MR<QDSIG[2]> = “0”                         : Sets the quadrature signals (Ph-A and Ph-B).

QEI0_IER<MAXEN[7]> = “1”                        : Enables the QEI0 MAX interrupt.
QEI0_MAX<QEIMAX[31:0]>                          : Enters a value in the QEI0 MAX counter.
= “00000000_00000000_00000001_10100000”          (arbitrary value: “0x1A0”)
QEI0_ISCR = “0x7FF”                             : Clears the QEI0 interrupt status flag.

: QEI0_CON<RESP[0]> = “1”                       : Resets the QEI0 position counter.

: QEI0_MR<QDMOD[0]> = “1”                       : Enables QEI0.
```

16 12-bit Analog-to-Digital Converter (ADC)

ADC block of A33M11x series consists of an independent ADC unit featuring the followings:

- 22 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 67 introduces pins assigned for ADC.

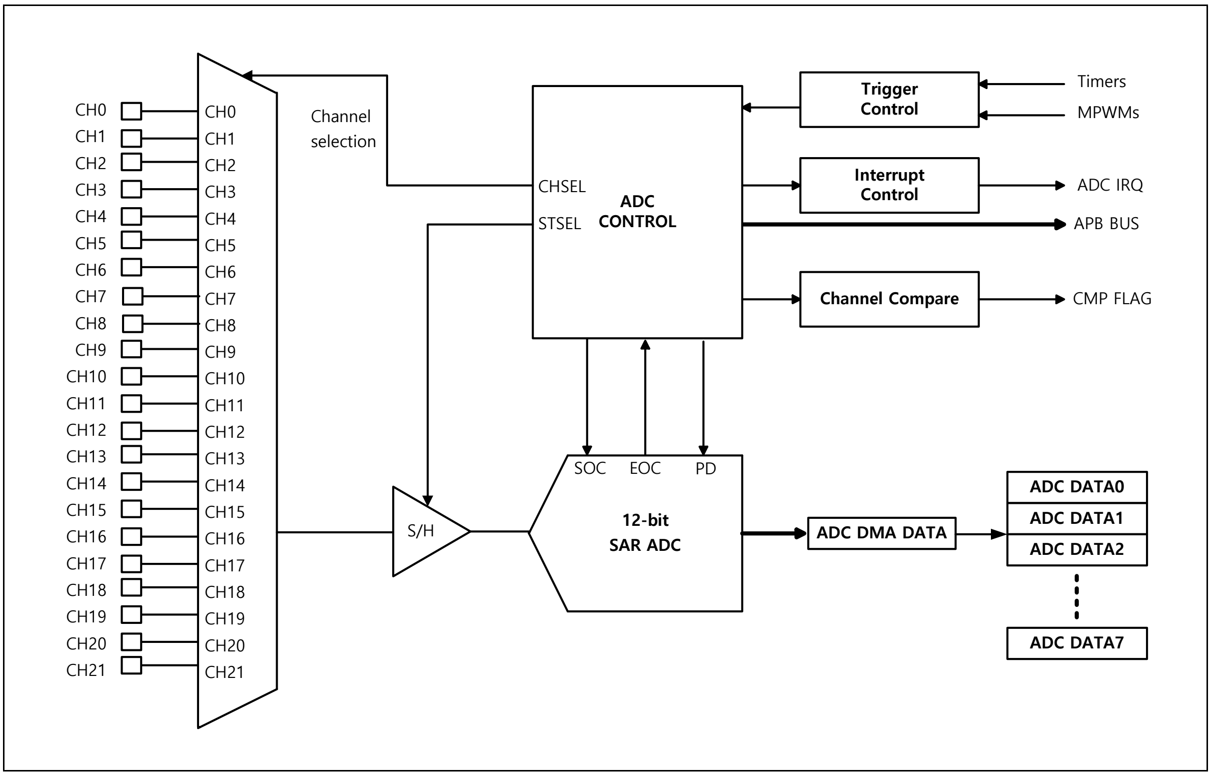
Table 67. Pin Assignment of ADC: External Pins

Pin name	Type	Description	Supported Packages		
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48)	**A33M114SN** (LQFP-44)
AN0	A	ADC input 0	O	O	O
AN1	A	ADC input 1	O	O	O
AN2	A	ADC input 2	O	O	O
AN3	A	ADC input 3	O	O	O
AN4	A	ADC input 4	O	O	X
AN5	A	ADC input 5	O	O	X
AN6	A	ADC input 6	O	O	O
AN7	A	ADC input 7	O	O	O
AN8	A	ADC input 8	O	O	O
AN9	A	ADC input 9	O	O	O
AN10	A	ADC input 10	O	O	O
AN11	A	ADC input 11	O	O	X
AN12	A	ADC input 12	O	O	X
AN13	A	ADC input 13	O	O	O
AN14	A	ADC input 14	O	O	O
AN15	A	ADC input 15	O	O	O
AN16	A	ADC input 16	X	O	O
AN17	A	ADC input 17	X	O	O
AN18	A	ADC input 18	X	O	O
AN19	A	ADC input 19	X	O	O

16.1 12-bit ADC block diagram

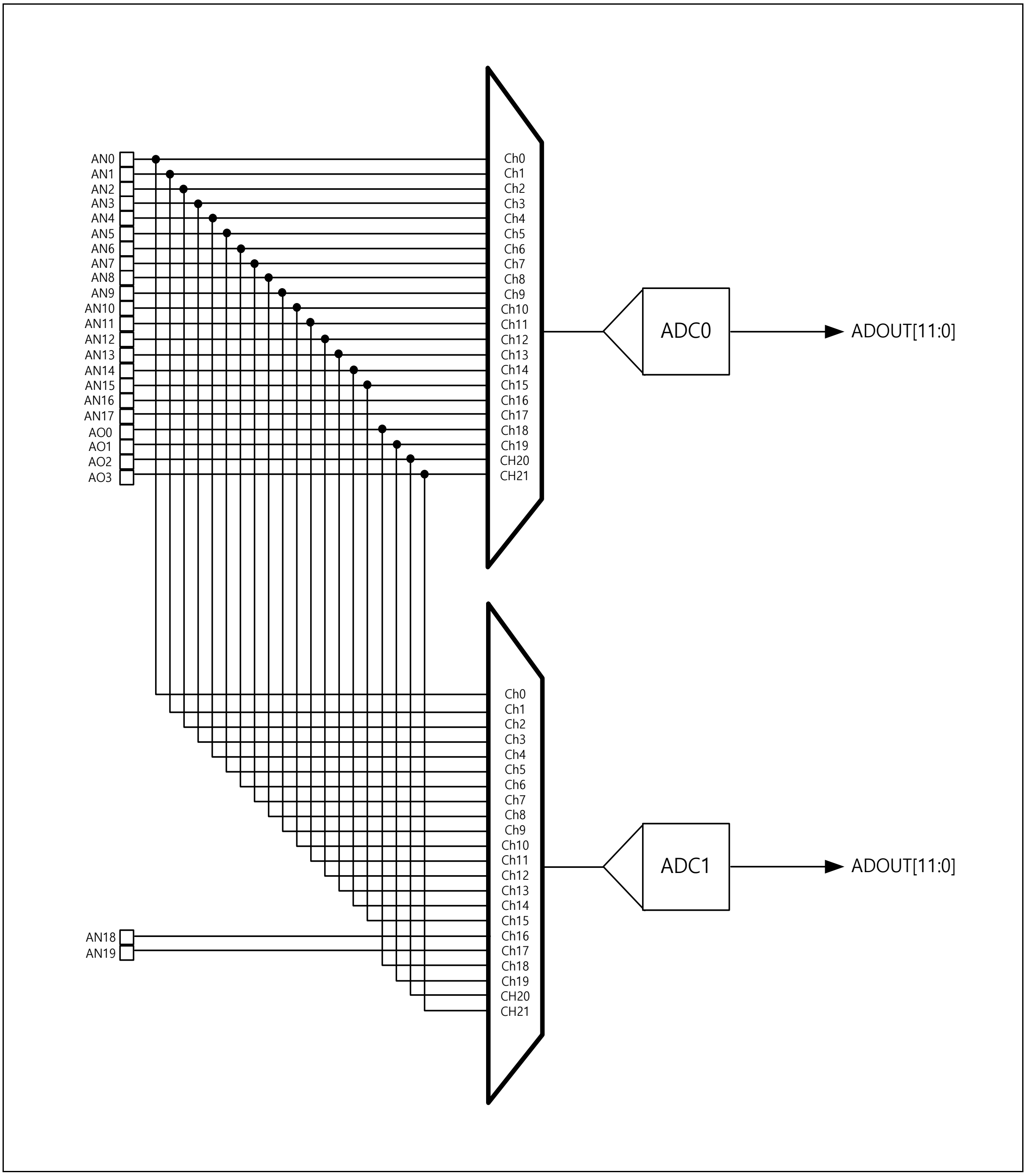
In this section, 12-bit ADC is described in a block diagram in Figure 132.

Figure 132. 12-bit ADC Block Diagram



16.2 Internal channel wiring

Figure 133. 12-bit ADC Internal Channel Wiring



16.3 Registers

Base address of ADC is introduced in the followings:

Table 68. Base Address of ADC

Name	Base address
ADC0	0x4000_B000
ADC1	0x4000_B100

Table 69. ADC Register Map

Name	Offset	Type	Description	Reset value	Reference
ADCn_MR	0x0000	RW	ADC n mode register	0x0000_0000	16.3.1
ADCn_CSCR	0x0004	RW	ADC n current sequence/channel register	0x0000_0000	16.3.2
ADCn_CCR	0x0008	RW	ADC n clock control register	0x0000_0080	16.3.3
ADCn_TRG	0x000C	RW	ADC n trigger select register	0x0000_0000	16.3.4
ADCn_SCSR1	0x0018	RW	ADC n channel select 1 register	0x0000_0000	16.3.5
ADCn_SCSR2	0x001C	RW	ADC n channel select 2 register	0x0000_0000	16.3.6
ADCn_CR	0x0020	RW	ADC n control register	0x0000_0000	16.3.7
ADCn_SR	0x0024	RC	ADC n status register	0x0000_0000	16.3.8
ADCn_IER	0x0028	RW	ADC n interrupt enable register	0x0000_0000	16.3.9
ADCn_DDR	0x002C	RO	ADC n DMA data register	0x0000_0000	16.3.10
ADCn_DR0	0x0030	RO	ADC n sequence 0 data register	0x0000_0000	16.3.11
ADCn_DR1	0x0034	RO	ADC n sequence 1 data register	0x0000_0000	16.3.11
ADCn_DR2	0x0038	RO	ADC n sequence 2 data register	0x0000_0000	16.3.11
ADCn_DR3	0x003C	RO	ADC n sequence 3 data register	0x0000_0000	16.3.11
ADCn_DR4	0x0040	RO	ADC n sequence 4 data register	0x0000_0000	16.3.11
ADCn_DR5	0x0044	RO	ADC n sequence 5 data register	0x0000_0000	16.3.11
ADCn_DR6	0x0048	RO	ADC n sequence 6 data register	0x0000_0000	16.3.11
ADCn_DR7	0x004C	RO	ADC n sequence 7 data register	0x0000_0000	16.3.11
ADCn_CMPR	0x0070	RW	ADC n channel compare register	0x0000_0000	16.3.12
ADCn_BCR	0x0074	RW	ADC n buffer control register	0x0000_0001	16.3.13

NOTE: n = 0 and 1.

16.3.1 ADCn_MR: ADC n mode register

ADCn_MR configures the modes of the ADC module. You should set this register prior to all other ADC registers according to the intended use of the ADC module.

ADC0_MR=0x4000_B000, ADC1_MR=0x4000_B100

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">Reserved</td><td>TRGINFO</td><td>CHINFO</td><td colspan="2">Reserved</td><td>DMAEN</td><td colspan="5">STSEL</td><td>Reserved</td><td colspan="3">SEQCNT</td><td>ADEN</td><td>ARST</td><td colspan="2">ADMOD</td><td colspan="2">Reserved</td><td colspan="2">TRGSEL</td></tr>
<tr><td colspan="10">-</td><td>0</td><td>0</td><td colspan="2">-</td><td>0</td><td colspan="5">00000</td><td>-</td><td colspan="3">000</td><td>0</td><td>0</td><td colspan="2">00</td><td colspan="2">-</td><td colspan="2">00</td></tr>
<tr><td colspan="10">-</td><td>RW</td><td>RW</td><td colspan="2">-</td><td>RW</td><td colspan="5">RW</td><td>-</td><td colspan="3">RW</td><td>RW</td><td>RW</td><td colspan="2">RW</td><td colspan="2">-</td><td colspan="2">RW</td></tr>
</table>

21	TRGINFO	Whether to enable or disable the trigger information option (in external trigger mode)			
		0	Disables the option.		
		1	Stores trigger source information in ADCnDR [31:24].		
20	CHINFO	Whether to enable or disable the channel information option			
		0	Disables the option.		
		1	ADCnDR [20:16] stores the information of the channel through which data conversion has been conducted.		
17	DMAEN	Whether to enable or disable DMA transfer (The bit must be set when ADC_MR.ADEN = 1)			
		If DMA is enabled, each DMA interrupt request is made when the ADC is signaled by the DMA controller that the previous DMA transfer has been completed. (The bit setting is valid in burst mode as well.)			
16 12	STSEL	Sampling time selection			
		The bits determine the time window in which the next trigger is recognized. This set value is applied right after the occurrence of the current trigger. ADC sampling time is calculated by (2 + STSEL [4:0]) MCLK cycles. The minimum sampling time is two MCLK cycles, and the sampling channel is always active when STSEL [4:0] = b'11111.			
10 8	SEQCNT	Number of conversions in a sequence If ADMOD [5:4] is 0 and SEQCNT [10:8] is not 0, the CSEQN value increments to the SEQCNT value by a trigger event. (The setting of the bit field is valid only in single/sequential modes.)			
		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	Whether to enable or disable the ADC module.			
		0	Disables.		
		1	Enables.		
6	ARST	Whether to stop or restart the ADC at the end of a sequence			
		0	Stops the ADC (ASTART must be set to 1 to restart).		
		1	Restarts the ADC at the end of a sequence.		
5 4	ADMOD	ADC mode selection			
		00	Single/sequential conversion modes		
		01	Burst conversion mode		
		10	Multiple conversion mode		
		11	Uses no modes.		
1 0	TRGSEL	Trigger selection			
		00	Disables event triggers and enables the soft trigger only.		

	01	Enables the timer event trigger and soft trigger.
	10	Enables the MPWM0 event trigger and soft trigger.
	11	Enables the MPWM1 event trigger and soft trigger.

NOTES:

1. If ADCMOD has been set for burst conversion mode, the ADC channels are assigned as SEQ0CH(BST0CH) through SEQ7CH(BST7CH). Burst mode always begins with SEQ0CH(BST0CH). (In three-burst mode, for example, the analog inputs to the SEQ0CH(BST0CH), SEQ1CH(BST1CH), and SEQ2CH(BST2CH) channels are converted in the order of channel number.)
2. If ADCMOD has been set for multiple mode, any trigger source set enabled in the TRG register triggers a start of conversion immediately when its triggering conditions are met, regardless of the sequence setting.

16.3.2 ADCn_CSCR: ADC n current sequence/channel register

ADCn_CSCR displays the ADC’s current sequence and channel. It is comprised of current sequence number bits and current active channel number bits. The CSEQN bits enable the user to instantly set the current sequence number. The register is 16 bits wide.

Before setting this register, you must set the ADEN bit in the ADCn_MR register.

ADC0_CSCR=0x4000_B004, ADC1_CSCR=0x4000_B104

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">Reserved</td><td colspan="3">CSEQN</td><td colspan="3">Reserved</td><td colspan="5">CACH</td></tr>
<tr><td colspan="21">-</td><td colspan="3">000</td><td colspan="3">-</td><td colspan="5">00000</td></tr>
<tr><td colspan="21">-</td><td colspan="3">RW</td><td colspan="3">-</td><td colspan="5">RO</td></tr>
</table>

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
		10000	ADC channel 16 is active
		10001	ADC channel 17 is active
		10010	ADC channel 18 is active
		10011	ADC channel 19 is active
		10100	ADC channel 20 is active
		10101	ADC channel 21 is active
		10110	ADC channel 22 is active
		10111	ADC channel 23 is active

Others	Reserved

16.3.3 ADCn_CCR: ADC n clock control register

ADCn_CCR controls the clock of the ADC module. This register is 16 bits wide.

ADC0_CCR=0x4000_B008, ADC1_CCR=0x4000_B108

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	EXTCLK	CLKINVT	Reserved
-	0	0000000	1	0	0	-
-	RW	RW	RW	RW	RW	-

15	ADCPDA	ADC disablement for power saving
		Does not set "1" here (it's optional bit)
14 8	CLKDIV	ADC clock division ratio (This value is valid when EXTCLK = 0)
		- 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 25MHz. (ADC Clock ≤ 25MHz)		
7	ADCPD	ADC power-down(stop) mode
		0 ADC normal mode
		1 ADC power-down(stop) mode
6	EXTCLK	ADC external clock configuration
		0 Internal clock (CLKDIV enablement)
		1 External clock (MCCR clock)
NOTE: In continuous conversion mode or burst conversion mode, the EXTCLK must be set to 0(internal clock).		
5	CLKINVT	Divide clock inversion (optional)
		0 Duty ratio of the divided clock is larger than 50%.
		1 Duty ratio of the divided clock is less than 50%.

16.3.4 ADCn_TRG: ADC n trigger select register

ADCn_TRG selects trigger sources for the ADC module.

ADC0_TRG=0x4000_B00C, ADC1_TRG=0x4000_B10C

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	3rd sequence trigger source
7 4	SEQTRG1	2nd sequence trigger source
3 0	SEQTRG0 BSTTRG	1st sequence trigger source Burst conversion trigger source

NOTE: In multiple mode, the 1st sequence is given the highest priority, and the 8th sequence is given the lowest priority. Table 70 shows trigger sources represented by each value.

Table 70. Trigger Source Table

Value	TIMER (TRGSEL[1:0]=0x1)	MPWM0 (TRGSEL[1:0]=0x2)	MPWM1 (TRGSEL[1:0]=0x3)
0	TIMER 0	MP0ATR1	MP1ATR1
1	TIMER 1	MP0ATR2	MP1ATR2
2	TIMER 2	MP0ATR3	MP1ATR3
3	TIMER 3	MP0ATR4	MP1ATR4
4	TIMER 4	MP0ATR5	MP1ATR5
5	TIMER 5	MP0ATR6	MP1ATR6
6	TIMER 6	PERIODU	PERIODU
7	TIMER 7	BOTTOMU	BOTTOMU
8		PERIODV	PERIODV
9		BOTTOMV	BOTTOMV
10		PERIODW	PERIODW
11		BOTTOMW	BOTTOMW
15	ASTART	ASTART	ASTART

NOTE: ASTART is a software trigger present in the CR register.

16.3.5 ADCn_SCSR1: ADC n channel select 1 register

ADCn_SCSR1 is the first register for the ADC module’s selection of channels. This register is 32 bits wide.

Each selected channel works based on the corresponding setting of the trigger select register. To write to this register, you must first set the ADEN bit enabled in the MR register.

ADC0_SCSR1=0x4000_B018, ADC1_SCSR1=0x4000_B118

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
-	00000	-	00000	-	00000	-	00000
-	RW	-	RW	-	RW	-	RW

28 24	SEQ3CH	4th conversion sequence channel selection
20 16	SEQ2CH	3rd conversion sequence channel selection
12 8	SEQ1CH	2nd conversion sequence channel selection
4 0	SEQ0CH	1st conversion sequence channel selection

NOTE: When setting the ADC mode to single mode, the channel must be set to SEQ0CH bits.

16.3.6 ADCn_SCSR2: ADC n channel select 2 register

ADCn_SCSR2 is the second register for the ADC module's selection of channels. This register is 32 bits wide.

Each selected channel works based on the corresponding setting of the trigger select register. To write to this register, you must first set the ADEN bit enabled in the MR register.

ADC0_SCSR2=0x4000_B01C, ADC1_SCSR2=0x4000_B11C

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
-	00000	-	00000	-	00000	-	00000
-	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

16.3.7 ADCn_CR: ADC n control register

ADCn_CR is used to control the ADC module.

ADC0_CR=0x4000_B020, ADC1_CR=0x4000_B120

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	Whether or not to stop ADC conversion	
		0	Has no effect.
		1	Stops ADC conversion (cleared at the next ADC clock cycle) If ASTOP sets enabled after a conversion cycle starts, the current conversion is completed.
1	TRGCLR	Option that clears all ADC trigger flags	
		0	Does not clear.
		1	Clears all trigger flags generated during the previous ADC operation.
0	ASTART	Whether or not to start ADC conversion	
		0	Does not start ADC conversion
		1	Starts ADC conversion (cleared at the next ADC clock cycle). To start ADC conversion, the ADCEN bit must be set to 1. If ARST is 0 in trigger event mode, setting ASTART to 1 will start ADC conversion, which then will be performed as many times as set with SEQCNT.

16.3.8 ADCn_SR: ADC n status register

ADCn_SR displays the status of the ADC.

ADC0_SR=0x4000_B024, ADC1_SR=0x4000_B124

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	DMAF	TRGIF	EOSIF	Reserved	EOCIF
-	0	-	0	0	0	-	0
-	RC	-	RO	RC	RC	-	RC

8	COMPIFLG	Compare interrupt flag	
		0	The interrupt has not occurred.
		1	The interrupt has occurred (Writing a 1 to the bit clears the flag).
4	DMAF	DMA done received flag (DMA transfer is completed)	
		0	Not flagged.
		1	Flagged.
3	TRGIF	ADC trigger interrupt flag	
		0	Not flagged.
		1	Flagged (Writing a 1 to bit clears the flag).
2	EOSIF	Sequence end interrupt flag	
		0	Not flagged.
		1	Flagged (Writing a 1 to bit clears the flag).
0	EOCIF	Sequence conversion end interrupt flag	
		0	Not flagged.
		1	Flagged (Writing a 1 to bit clears the flag).

NOTE: To poll the flag, you must use the EOCIF bit. Following is the example code:

```
while ( (ADCn_SR & 0x01) != 1 );              // EOCIF Flag Checking

  ADCn_SR = 0x1;                               // EOCIF Flag Clear
```

16.3.9 ADCn_IER: ADC n interrupt enable register

ADCn_IER determines the enablement of ADC interrupts.

ADC0_IER=0x4000_B028, ADC1_IER=0x4000_B128

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	Whether to enable or disable the DMA done interrupt.	
		0	Disables.
		1	Enables.
3	TRGIE	Whether to enable or disable the ADC trigger conversion interrupt	
		0	Disables.
		1	Enables.
2	EOSIE	Whether to enable or disable the ADC sequence conversion interrupt	
		0	Disables.
		1	Enables.
0	EOCIE	Whether to enable or disable the ADC single conversion interrupt	
		0	Disables.
		1	Enables.
NOTE: Burst mode sets the EOSIE bit to 1 and check the EOSIF bit in the ADCn_SR register.			

16.3.10 ADCn_DDR: ADC n DMA data register

ADCn_DDR manages the ADC module's DMA data. It displays the results of the ADC module's DMA conversions.

ADC0_DDR=0x4000_B02C, ADC1_DDR=0x4000_B12C

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	0x000	-
RO	RO	RO	RO	RO	RO	RO	RO	-	RO	RO	-

31 24	TRGINFOx (x=0–7)	ADC trigger information (Indicates whether or not each trigger source has been captured until the end of conversion (EOC).) * To use this bit field, you must set the TRGINFO bit enabled in the MR register. For multiple mode: The lower the TRGINFO number, the higher priority it has. If a trigger is pending due to another ongoing trigger, multiple TRGINFO bits can be read as 1. For single/sequential modes: The bit field displays which trigger sources are pending due to another ongoing trigger. You can find out which trigger source is being currently processed by referring to the CSEQN bit field of the CSCR register.
20 16	ADMACH	DMA ADC channel indicator * To use this bit field, you must set the CHINFO bit enabled in the MR register.
15 4	ADDMAR	DMA ADC conversion result data (12-bit)

NOTE: Even when DMA is inactive, data is temporarily stored in this register before stored in the designated buffer. Additionally, the register also displays which channel the data belongs to.

16.3.11 ADCn_DR: ADC n sequence 0 to 7 data register

ADCn_DR displays the results of ADC conversions. There are eight of these registers, which of each represents one sequence.

ADC0_DR0=0x4000_B030, ADC0_DR1=0x4000_B034, ADC0_DR2=0x4000_B038, ADC0_DR3=0x4000_B03C, ADC0_DR4=0x4000_B040, ADC0_DR5=0x4000_B044, ADC0_DR6=0x4000_B048, ADC0_DR7=0x4000_B04C, ADC1_DR0=0x4000_B130, ADC1_DR1=0x4000_B134, ADC1_DR2=0x4000_B138, ADC1_DR3=0x4000_B13C ADC1_DR4=0x4000_B140, ADC1_DR5=0x4000_B144, ADC1_DR6=0x4000_B148, ADC1_DR7=0x4000_B14C

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 to 7)	ADC trigger information (Indicates whether or not each trigger source has been captured until the end of conversion (EOC).) * To use this bit field, you must set the TRGINFO bit enabled in the MR register. For multiple mode: The lower the TRGINFO number, the higher priority it has. If a trigger is pending due to another ongoing trigger, multiple TRGINFO bits can be read as 1. For single/sequential modes: The bit field displays which trigger sources are pending due to another ongoing trigger. You can find out which trigger source is being currently processed by referring to the CSEQN bit field of the CSCR register.
20 16	ACH	ADC channel information * To use this bit field, you must set the CHINFO bit enabled in the MR register.
15 4	ADDATA	ADC input data

16.3.12 ADCn_CMPR: ADC n channel compare register

ADCn_CMPR controls comparison between ADC channels.

ADC0_CMPR=0x4000_B070, ADC1_CMPR=0x4000_B170

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	Whether to enable or disable the compare interrupt	
		0	Disables.
		1	Enables.
23	COMPEN	Whether to enable or disable the compare operation	
		0	Disables.
		1	Enables.
21	LTE	AD conversion value output timing setting	
		0	The ADC value is larger than the CVAL value.
		1	The ADC value is smaller than the CVAL value.
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.
		10001	Compare channel is ADC channel 17.
		10010	Compare channel is ADC channel 18.
		10011	Compare channel is ADC channel 19.
		10100	Compare channel is ADC channel 20.
		10101	Compare channel is ADC channel 21.
		Others	Reserved
15 4	CVAL	Compare value	

16.3.13 ADCn_BCR: ADC n buffer control register

ADCn_CMPR controls comparison between ADC channels.

ADC0_BCR=0x4000_B074, ADC1_BCR=0x4000_B174

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	BYPSEL	Reserved	BUFEN	BYPSEL
-	000	-	0	1
-	RW	-	RW	RW

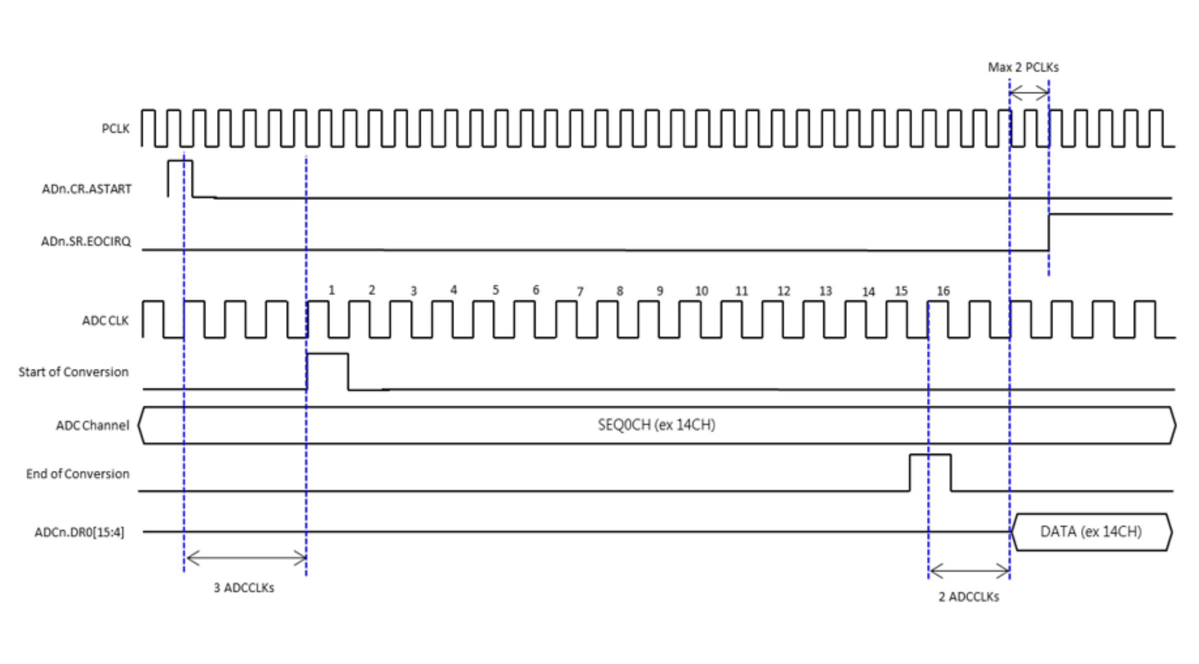
6 4	TBUFSEL	Buffering Off Time Selection bit	
		Can Select 1*MCLK ~ 8*MCLK	
1	BUFEN	ADC Input Buffer Enable bit	
		0	Input Buffer Power Down Period
		1	Input Buffer Operation Period
0	BYPSEL	ADC Input Buffer Bypass Selection bit	
		0	Input Buffer Operation mode
		1	Input Buffer Bypass mode

16.4 Functional description

16.4.1 ADC single mode timing diagram

When both MR.ADMOD and MR.SEQCNT are set to 0x0, ADC conversion begins by setting the CR.ASTART bit to 1. The start of conversion (SOC) becomes active three ADC clock cycles after the enablement of CR.ASTART. And once the end of conversion (EOC) becomes active, ADC_SR.EOC becomes enabled after two PCLK cycles and then two ADC clock cycles.

Figure 134. ADC Single Mode Timing: When ADCn.MR.AMOD = 0



16.4.2 ADC burst mode timing diagram

There are two types of SOC trigger sources in burst mode: TRG events (timers and MPWM) and ASTART. If TRGSEL is set for the timer or MPWM event triggers, SOCs are triggered by the TRG.BSTTRG's set trigger.

If TRG.BSTTRG is set to TIMER 3, for example, ADC conversion gets started by the TIMER 3 trigger. Once an event is triggered by BSTTRG's set trigger, the ADC shifts the ADC channels as many times as set in MR.SEQCNT. Refer to Figure 136.

Figure 135. ADC Burst Mode Timing: When ADCn.MR.ADMOD = 1

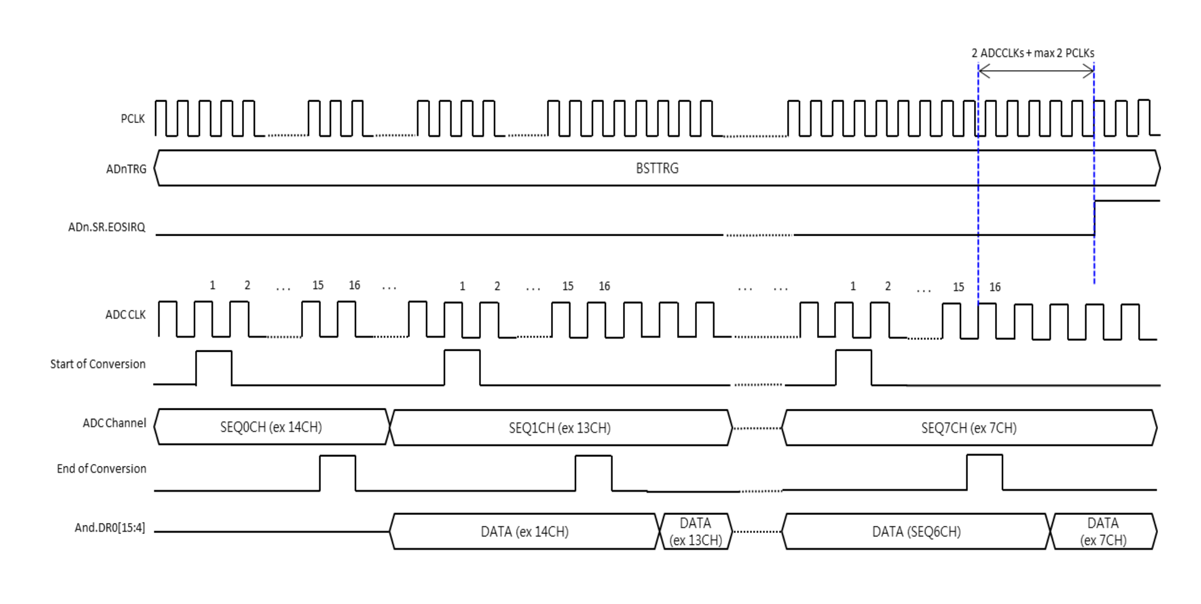
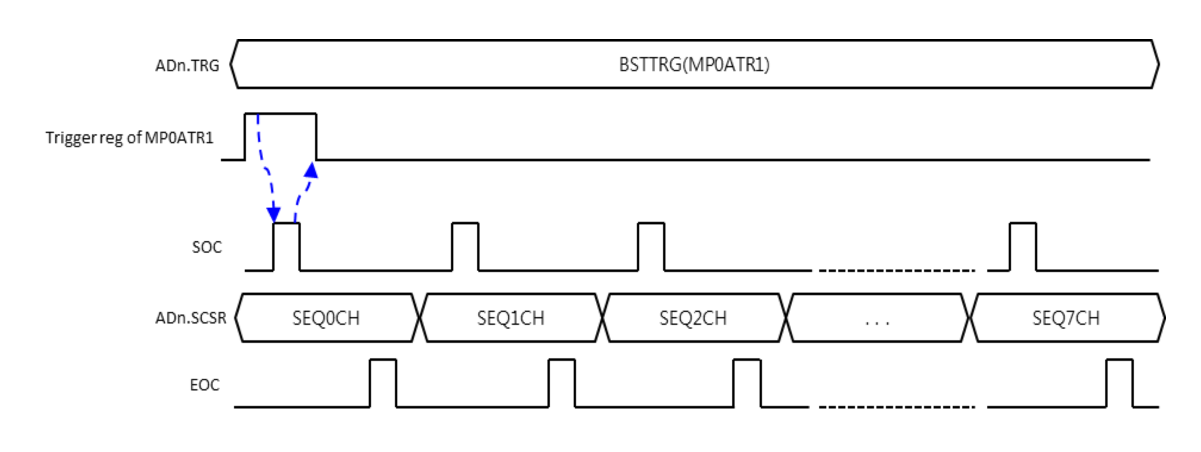


Figure 136. ADC Trigger Timing in Burst Mode: SEQCNT = 3'b111, 8 Sequential Conversion



16.4.3 ADC sequential mode timing diagram

To enter sequential mode, MR.ADMOD must be set to 2'b00, and MR.SEQCNT must be set to a value other than 3'b000.

Operations in sequential mode are almost identical to those in burst mode. The difference is SOC trigger sources. In sequential mode, each sequence's SOC is triggered by the corresponding SEQTRGx trigger, followed by as many conversions as defined by SEQCNT. Refer to Figure 138.

Figure 137. ADC Sequential Mode Timing: When MR.AMOD = 0 and MR.SEQCNT ≠ 0

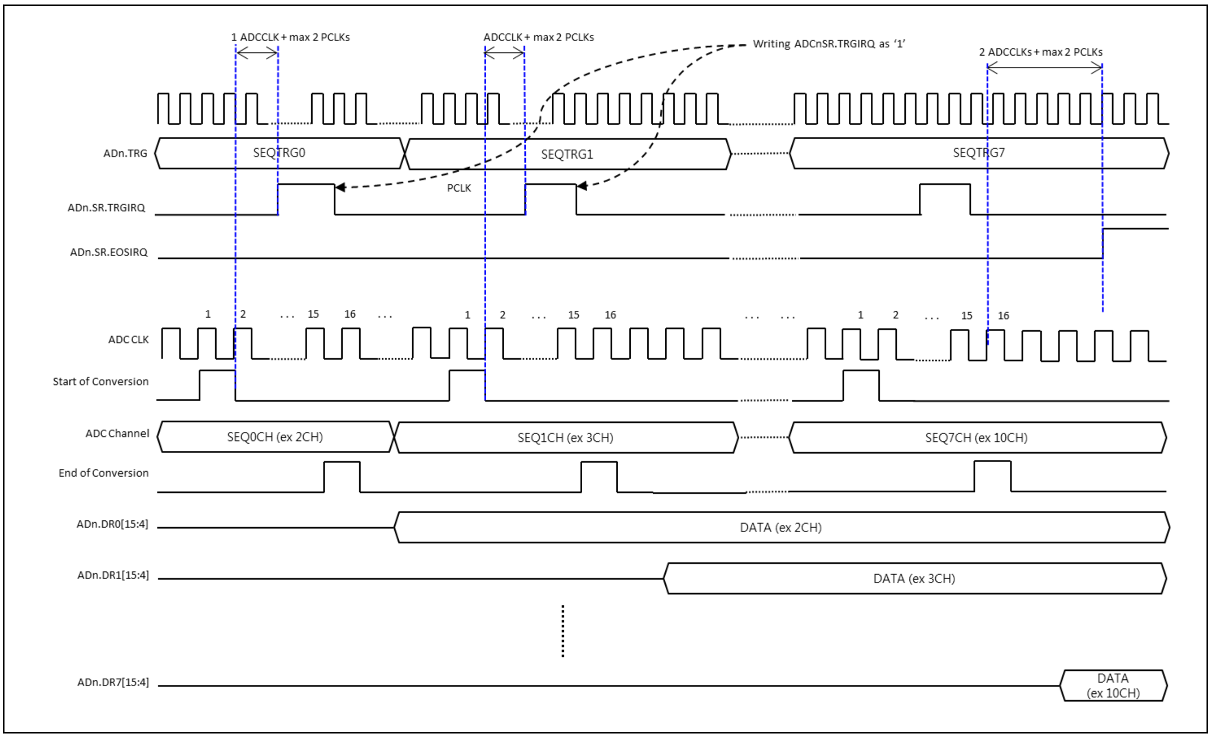
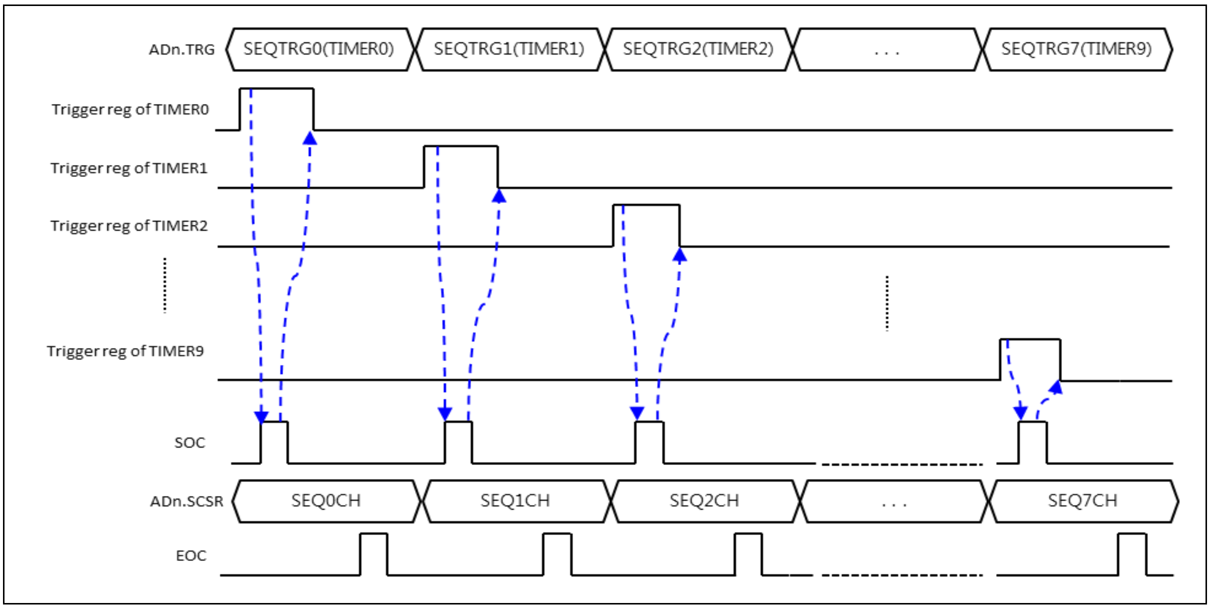


Figure 138. ADC Trigger Timing in Sequential Mode: SEQCNT = 3'b111, 8 Sequential Conversion



16.4.4 ADC multiple mode timing diagram

To enter multiple mode, MR.ADMOD must be set to 2'b10, and MR.SEQCNT must be set to a value other than 3'b000.

You can set your desired trigger sources in TRG.SEQTRG. Each of these sources triggers an SOC when the triggering conditions are met, regardless of the sequence setting in SEQTRG. For example, if MR.SEQCNT is set to 3’b011 and four trigger sources are selected from among the timer and MPWM trigger sources, setting CR.ASTART enabled lets conversions triggered by these trigger sources in the order of reception, unlike sequential or burst modes.

Figure 139. ADC Multiple Mode Timing: When MR.AMOD = 2 and MR.SEQCNT ≠ 0

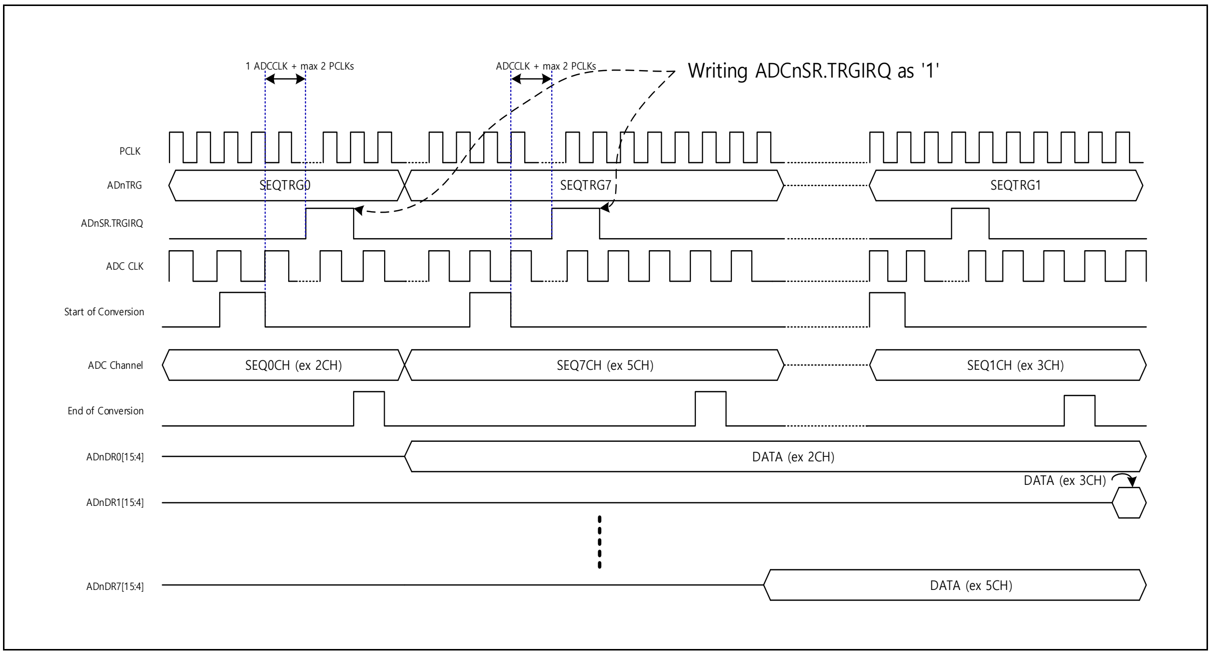
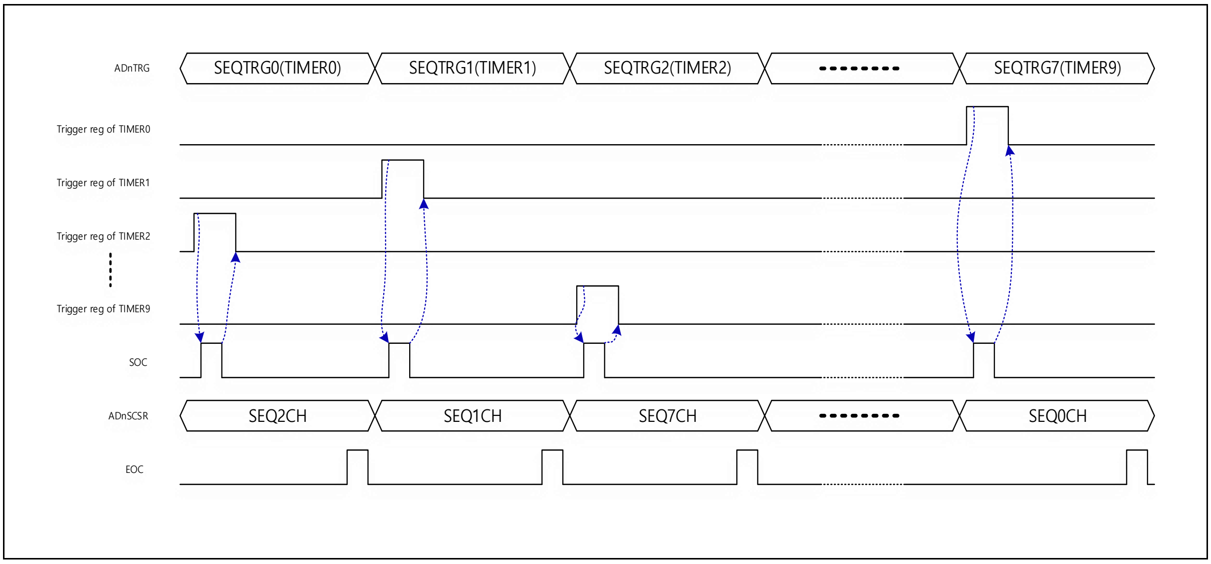


Figure 140. ADC Trigger Timing in Multiple Mode: SEQCNT = 3’b111, 8 Multiple Conversion



16.4.5 Setting examples

<Example 1> Initial configuration of ADC0

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”              : Enables SCU register access.
SCU_PER2<ADC0[20]> = “1”                      : Enables the ADC peripheral.
SCU_PCER2<ADC0[20]> = “1”                     : Enables the ADC peripheral clock.
PORTEN<PORTEN[7:0]> = “0x15”
PORTEN<PORTEN[7:0] = “0x51”                   : Enables PCU register access.
PA_MR1<P0[3:0]> = 0x7;                        : Configures the AN0 port.
PORTEN<PORTEN[7:0] = “0”                      : Disables PCU register access
ADC_MR<TRGINFO[21]> = “1”                     : Enables the trigger information function.
ADC_MR<CHINFO[20]> = “1”                      : Enables the channel information function.
ADC_MR<DMAEN[17]> = “0”                       : Disables DMA.
ADC_MR<STSEL[16:12]> = “1”                    : Sets the sampling time to 1.
ADC_MR<SEQCNT[10:8]> = “0”                    : Sets the number of conversions to 1.
ADC_MR<ADEN[7]> = “1”                         : Enables the ADC block.
ADC_MR<ARST[6]> = “0”                         : Disables conversion restarting at the end of conversion.
ADC_MR<ADMOD[5:4]> = “00”                     : Sets the ADC to single mode.
ADC_MR<TRGSEL[1:0]> = “0”                     : Selects “software trigger only” as a trigger source.
```

<Example 2> DMA channel configuration

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”              : Enables SCU register access.
 SCU_PER1<DMA[4]> = “1”                       : Enables the DMA peripheral.
 SCU_PCER1<DMA[4]> = “1”                      : Enables the DMA peripheral clock.
 DMA_CR<TRANSCNT[27:16]> = “0x001”            : Sets the number of DMA transfers to 1.
 DMA_CR<PERISEL[12:8]> = “0x13”               : Selects ADC0 as the peripheral for DMA transfer.
 DMA_CR<SIZE[3:2]> = “10”                     : Sets the DMA transfer size to word size.
 DMA_CR<DIR[1]> = “1”                         : Selects DMA Rx transfer.
```

<Example 3> ADC conversion enablement

```
ADC_CSCR1<SEQ0CH[4:0]> = “0”                  : Selects channel 0.
ADC_CR<ASTART[0]> = “1”                       : Starts ADC conversion.
while (ADC_SR<EOCIF[0]> == 0);                : Checks for the EOC flag.
adc_data = ADC_DR;                            : Stores the ADC data.
ADC_SR<EOCIF[0]> = 1;}                        : Clears the EOC flag.
```

17 Analog Front End (AFE)

Analog Front End (AFE) is an Op-Amp and a comparator interface controller. A33M11x series is equipped with four Op-Amps and comparators. The OPAMPs amplify their voltage difference between positive analog input signals and negative analog input signals. A comparator outputs a signal from its I/O pin or triggers an interrupt based on comparison between the voltages of two analog signals.

The AFE of A33M11x series features the followings:

- Four op-amps and comparators
- OPAMP outputs are used in connection with ADC channels or AOx Pins
- The comparator output supports the debounce function.
- Level and edge interrupt support

Table 71 introduces pins assigned for AFE.

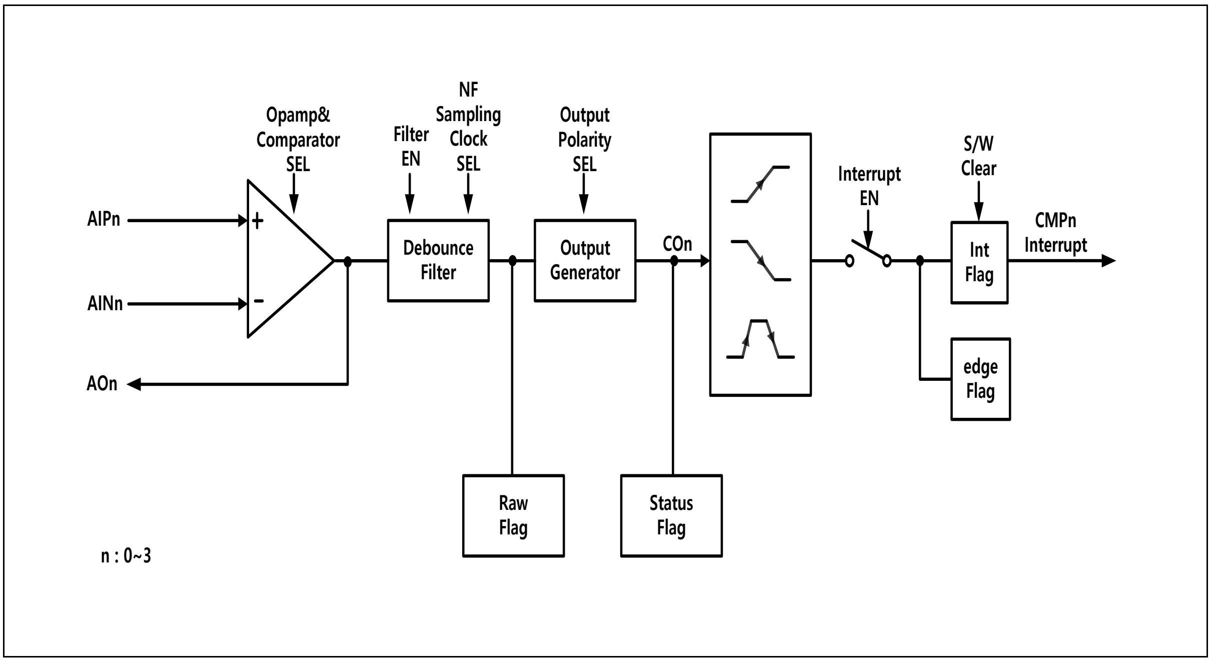
Table 71. Pin Assignment of AFE: External Pins

Pin name	Type	Description	Supported Packages	
			A33M116RL **A33M116RM** **A33M114RL** (LQFP-64)	**A33M116CL** **A33M114CL** **A33M114SN** (LQFP-48, 44)
AIN0	A	Analog Input 0 (-)	O	O
AIP0	A	Analog Input 0 (+)	O	O
AO0/CO0	A	Op-Amp & Comparator Output 0	O	O
AIN1	A	Analog Input 1 (-)	O	O
AIP1	A	Analog Input 1 (+)	O	O
AO1/CO1	A	Op-Amp & Comparator Output 1	O	O
AIN2	A	Analog Input 2 (-)	O	O
AIP2	A	Analog Input 2 (+)	O	O
AO2/CO2	A	Op-Amp & Comparator Output 2	O	O
AIN3	A	Analog Input 3 (-)	O	O
AIP3	A	Analog Input 3 (+)	O	O
AO3/CO3	A	Op-Amp & Comparator Output 3	O	O

17.1 AFE block diagram

In this section, AFE is described in a block diagram in Figure 141.

Figure 141. AFE Block Diagram



17.2 Registers

Base address of AFE is introduced in the followings:

Table 72. Base Address of AFE

Name	Base address
AFE	0x4000_B300

Table 73. AFE Register Map

Name	Offset	Type	Description	Reset value	Reference
AFE0_CR	0x0000	RW	AFE0 control register	0x0000_0000	17.2.1
AFE1_CR	0x0004	RW	AFE1 control register	0x0000_0000	17.2.1
AFE2_CR	0x0008	RW	AFE2 control register	0x0000_0000	17.2.1
AFE3_CR	0x000C	RW	AFE3 control register	0x0000_0000	17.2.1
CMP_DBR	0x0020	RW	AFE comparator debounce register	0x0000_0000	17.2.2
CMP_ICR	0x0024	RW	AFE comparator interrupt control register	0x0000_0000	17.2.3
CMP_SR	0x0028	RC	AFE comparator status register	0x0000_0000	17.2.4

NOTE: n = 0, 1 and 2

17.2.1 AFEn_CR: AFE n control register

This register is used to control the OPAMP and COMPARATOR functions of the AFE module. The four registers (AFE0_CR, AFE1_CR, AFE2_CR, and AFE3_CR) have the same function, so the bit positions are the same.

AFE0_CR=0x4000_B300, AFE1_CR=0x4000_B304
AFE2_CR=0x4000_B308, AFE3_CR=0x4000_B30C

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	CMPOSEL	Reserved	OPAEN	CMPIEN	UGAINEN	CMPEN
-	0	-	0	0	0	0
-	RW	-	RW	RW	RW	RW

15	CMPOSEL	Comparator output polarity	
		0	General output.
		1	General output inversion
3	OPAEN	Whether to enable or disable the Op-Amp	
		0	Disables
		1	Enables
2	CMPIEN	Whether to enable or disable the comparator interrupt	
		0	Disables
		1	Enables
1	UGAINEN	Whether to enable or disable unit gain (Enabling the bit dismisses the GAINSEL value, gain = 1)	
		0	Disables
		1	Enables
0	CMPEN	Whether to enable or disable the comparator	
		0	Disables
		1	Enables

NOTES:

1. If both OPAMPEN [3] and CMPEN [0] bits are enabled, only the comparator will operate.
2. For the analog outputs, the PxMux bits of the Pn_MR1 and Pn_MR2 registers must be set to b'111 (Function 4) and the Px bit of the Pn_CR register must be set to b'1x (input).
   If the Px bit of the Pn_CR register is set to b'00 (push-pull output) or b'01 (open-drain output), the port may not operate correctly.
   Analog output pins are listed below:
   PA0(AO0), PA7(AO1), PA8(AO2), PA15(AO3), PC13(XOUT)

17.2.2 CMP_DBR: comparator debounce register

This register is set to use the De-bounce function when using the comparator of the AFE module.

CMP_DBR=0x4000_B320

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	C3FLTSEL	C2FLTSEL	C1FLTSEL	C0FLTSEL
-	0x0	0x0	0x0	0x0
-	RW	RW	RW	RW

15 12	C3FLTSEL	Comparator 3 filter counter bits selection Filter value by FLTSEL X (1/PCLK) If PCLK is 1MHz and FLTSEL is 2, the signal must remain at least 2us before a flag is generated.
11 8	C2FLTSEL	Comparator 2 filter counter bits selection Filter value by FLTSEL X (1/PCLK) If PCLK is 1MHz and FLTSEL is 2, the signal must remain at least 2us before a flag is generated.
7 4	C1FLTSEL	Comparator 1 filter counter bits selection Filter value by FLTSEL X (1/PCLK) If PCLK is 1MHz and FLTSEL is 2, the signal must remain at least 2us before a flag is generated.
3 0	C0FLTSEL	Comparator 0 filter counter bits selection Filter value by FLTSEL X (1/PCLK) If PCLK is 1MHz and FLTSEL is 2, the signal must remain at least 2us before a flag is generated.

17.2.3 CMP_ICR: comparator interrupt control register

This register is the comparator 0/1/2/3 interrupt control register of the AFE module.

CMP_ICR=0x4000_B324

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	INTPOL3	Reserved	INTTYPE3	INTPOL2	Reserved	INTTYPE2	INTPOL1	Reserved	INTTYPE1	INTPOL0	Reserved	INTTYPE0
-	0	-	0	0	-	0	0	-	0	0	-	0
-	RW	-	RW	RW	-	RW	RW	-	RW	RW	-	RW

4n+3	INTPOLn (n=0~3)	Interrupt polarity selection (The setting of this bit is ignored when “both edges” is selected as the interrupt type)	
		0	Low (falling)
		1	High (rising)
4n+1	INTTYPEn (n=0~3)	Interrupt type selection	
		00	Disable
		01	Level
		10	Single edge
		11	Both edges

17.2.4 CMP_SR: comparator status register

This register is a register that indicates the operation status of the Comparator 0/1/2/3 of the AFE module.

CMP_SR=0x4000_B328

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		CMP3EOPL	CMP3RFLAG	CMP3INTF	CMP3FLAG	CMP2EOPL	CMP2RFLAG	CMP2INTF	CMP2FLAG	CMP1EOPL	CMP1RFLAG	CMP1INTF	CMP1FLAG	CMP0EOPL	CMP0RFLAG	CMP0INTF	CMP0FLAG
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RC	RO	RC	RO	RC	RO	RC	RO	RC	RO	RC	RO	RC	RO	RC	RO

4n+3	COMPnEPOL (n=0~3)	The edge polarity status of comparator n in the comparison state (Only used in both edge mode)	
		0	Falling Edge
		1	Rising Edge
4n+2	COMnRFLAG (n=0~3)	Comparator raw flag	
		0	Low.
		1	High.
4n+1	COMPnINTF (n=0~3)	Comparator interrupt flag	
		0	Not occurred.
		1	Occurred (Cleared when writing to the bit).
4n	COMPnFLAG (n=0~3)	Comparator output flag	
		0	Not occurred.
		1	Occurred.

NOTE: The CMPnEPOL bit is only available in both edge interrupt modes (not used polling mode). In single edge interrupt mode, it is fixed to ‘1’ or ‘0’ according to the INPOLn bit setting in CMP_ICR.

17.3 Functional description

The OPAMPs amplify their voltage difference between positive analog input signals and negative analog input signals. The OPAMPs are connected to ADC channels 18, 19, 20, and 21, respectively. These cannot be changed to other ADC channels. Since signals from these channels can be amplified and fed to ADC0 and ADC1, they can be used for a wide range of applications.

The comparator compares the voltages of the input signal and the reference signal and then, based on the comparison result, generates a digital output signal. If the reference voltage is lower than the input voltage, the digital output signal value is 0; otherwise, the signal value is 1. The comparator is enabled by writing a 1 to the CMPEN bit of the AFEn_CR register. The output value of the comparator can be inverted by the CMPOSEL bit in the AFEn_CR register. The output value before inversion can be checked with the CMPnRFLAG bit in the CMP_SR register, and the output value after inversion can be checked with the CMPnFLAG bit.

If the interrupt has been enabled, you can check for its occurrence at the CMPnINTF bit.

17.3.1 Op-Amp configuration

Each OPAMP’s gain value can be configured by setting the positive analog input signal and the negative analog input signal. The unit gain function is also supported, which allows you to set the gain value to 1.

17.3.2 Comparator reference and input voltage configuration

The comparator reference voltage can be set by inputting the negative analog input voltage to the external I / O pin, and the comparator input voltage can be set by inputting positive analog input voltage to the external I / O pin. The comparator module of the A33M11x Series provides the debounce filter function. This function sets the CnFLTSEL value in the CMP_DBR register to filter the noise that interferes with the input port, thereby filtering out the normal signal.

17.3.3 Comparator interrupt function

The comparator interrupt occurs at every rising or falling edge of the comparator output. When the INTPOLn bit in the CMP_ICR register is set to '1' after activating the CMPIEN bit in the AFEn_CR register to '1', the rising edge is detected, and the falling edge is detected when it is set to '0'. You can check the status of this interrupt at the CMPnINTF bit of the CMP_SR register. Writing a 1 to the bit clears the interrupt flag. Note that the CMPnEPOL bit can only be used in both edge modes. In single edge mode, it is fixed to '1' or '0' depending on the INTPOLn bit setting in CMP_ICR.

17.3.4 Setting examples

<Example 1> Op-Amp configuration

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”                  : Enables SCU register access.
SCU_PER2<AFE[24]> = “1”                           : Enables the AFE peripheral.
SCU_PCER2<AFE[24]> = “1”                          : Enables the AFE peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                   : Enables PORTEN (enter 0x15 and then 0x51).

PA_MR1<P0MUX[2:0]> = “111”                        : Sets Port A pin 0 as AO0.
PA_MR1<P1MUX[6:4]> = “111”                        : Sets Port A pin 1 as AIN0
PA_MR1<P2MUX[10:8]> = “111”                       : Sets Port A pin 2 as AIP0.
PA_PRCR<PUE0[1:0]> = “00”                         : Disables pull-up/pull-down at Port A pin 0.
PA_PRCR<PUE1[3:2]> = “00”                         : Disables pull-up/pull-down at Port A pin 1.
PA_PRCR<PUE2[5:4]> = “00”                         : Disables pull-up/pull-down at Port A pin 2.

AFE0_CR<AMPISEL[16]> = “1”                        : Sets the opamp current to 17 percent.
AFE0_CR<OPAEN[3]> = “1”                           : Enables the opamp.
```

<Example 2> Comparator0 configuration

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”                  : Enables SCU register access
SCU_PER2<AFE[24]> = “1”                           : Enables the AFE peripheral.
SCU_PCER2<AFE[24]> = “1”                          : Enables the AFE peripheral clock.

PCU_PORTEN<PORTEN[7:0]> = “0x15”
PCU_PORTEN<PORTEN[7:0] = “0x51”                   : Enables PORTEN (enter 0x15 and then 0x51

PA_MR1<P0MUX[2:0]> = “111”                        : Sets Port A pin 0 as AO0.
PA_MR1<P1MUX[6:4]> = “111”                        : Sets Port A pin 1 as AIN0
PA_MR1<P2MUX[10:8]> = “111”                       : Sets Port A pin 2 as AIP0.
PA_PRCR<PUE0[1:0]> = “00”                         : Disables pull-up/pull-down at Port A pin 0.
PA_PRCR<PUE1[3:2]> = “00”                         : Disables pull-up/pull-down at Port A pin 1.
PA_PRCR<PUE2[5:4]> = “00”                         : Disables pull-up/pull-down at Port A pin 2.

CMP_DBR<C0FLTSEL[3:0]> = “1010”                   : Sets the comparator0 debounce filter to 10
AFE0_CR<CMPEN[0]> = “1”                           : Enables the comparator.
```

18 Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check (CRC) module is used to load 32/16/8/7-bit CRC codes. Application programs employs CRC-based technologies to examine the integrity of data transfers, storages, and Flash memories in conformance with functional safety standards.

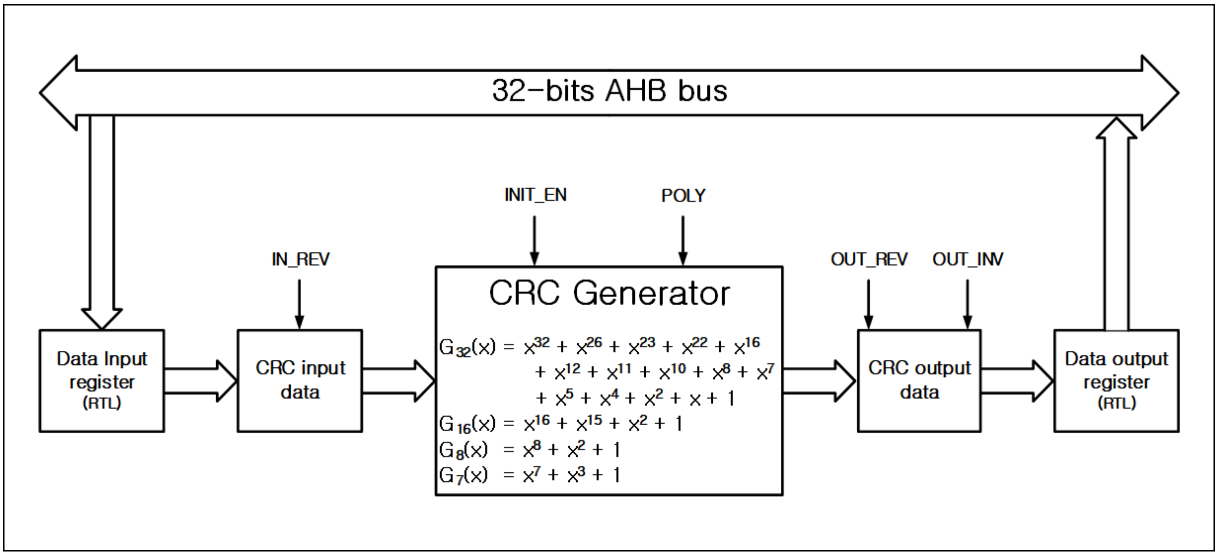
The CRC of A33M11x series features the followings:

- Automatic CRC and user CRC modes
- Handles 8-, 32-bit data size
- Input buffer to avoid bus stall during calculation
- CRC computation done in 4 AHB clock cycles (HCLK) for the 32-bit data size
- Reversibility option on I/O data

18.1 CRC block diagram

In this section, CRC is described in a block diagram in Figure 142.

Figure 142. CRC Block Diagram


32-bits AHB bus
INIT_EN
POLY
IN_REV
OUT_REV
OUT_INV
CRC Generator
$G_{32}(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
$G_{16}(x) = x^{16} + x^{15} + x^{2} + 1$
$G_{8}(x) = x^{8} + x^{2} + 1$
$G_{7}(x) = x^{7} + x^{3} + 1$
Data Input register (RTL)
CRC input data
CRC output data
Data output register (RTL)

18.2 Registers

Base address of CRC is introduced in the followings:

Table 74. Base Address of CRC

Name	Base address
CRC	0x4100_2000

Table 75. CRC Register Map

Name	Offset	Type	Description	Reset value	Reference
CRC_CTRL	0x0000	RW	CRC control register	0x0000_0000	18.2.1
CRC_INIT	0x0004	RW	CRC initial value register	0xFFFF_FFFF	18.2.2
CRC_IDR	0x0008	WO	CRC input data register	0x0000_0000	18.2.3
CRC_ODR	0x0008	RO	CRC output data register	0xFFFF_FFFF	18.2.4
CRC_STAT	0x000C	RW	CRC status register	0x0000_0000	18.2.5

18.2.1 CRC_CTRL: CRC control register

CRC_CTRL is used to control the CRC module.

CRC_CTRL=0x4100_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	OUT_INV	OUT_REV	Reserved	IN_REV	Reserved	DMADINT	Reserved	POLY	INIT_EN
-	0	0	-	0	-	0	-	00	0
-	RW	RW	-	RW	-	RW	-	RW	WO

21	OUT_INV	Whether to enable or disable CRC output data inversion	
		0	Disables.
		1	Enables.
20	OUT_REV	Whether to enable or disable CRC output data reverse	
		0	Disables.
		1	Enables.
16	IN_REV	Input data reverse mode selection	
		0	Does not reverse the input data.
		1	Reverses the input data.
8	DMADINT	Whether to enable or disable the DMA done interrupt	
		0	Disables.
		1	Enables.
2 1	POLY	Polynomial selection	
		00	CRC32 (0x04C1_1DB7)
		01	CRC16 (0x8005)
		10	CRC8 (0x07)
		11	CRC7 (0x09)
0	INIT_EN	Whether or not to apply the CRC initial value register	
		0	No Effect
		1	Applies the initial value register value.

18.2.2 CRC_INIT: CRC initial value register

The CRC initial value is written to CRC_INIT.

CRC_INIT=0x4100_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
INIT			
0xFFFF_FFFF			
RW			

31 0	INIT	CRC initial value
NOTE:	To write INIT data to the CRC register, the CTRL register's INIT_EN bit must be enabled. For example, writing INIT = "0x8005" and writing a 1 to CTRL's zeroth bit (INIT_EN) changes the CRC16 polynomial value to 8005; once this value is assigned to the CRC_RLT register, the calculation result is re-written to this register.	

18.2.3 CRC_IDR: CRC input data register

CRC Input Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

CRC_IDR=0x4100_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
INPUT			
0x0000_0000			
WO			

31 0	DATAIOD	CRC input data
		Once data is entered in this bit field, its polynomial result is automatically written at CRC_ODR register

18.2.4 CRC_ODR: CRC output data register

CRC Output Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

CRC_ODR=0x4100_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
OUTPUT			
0xFFFF_FFFF			
RO			

31 0	OUTPUT	CRC output data

18.2.5 CRC_STAT: CRC status register

CRC_STAT displays the operating status of CRC.

CRC_STAT = 0x4100_200C

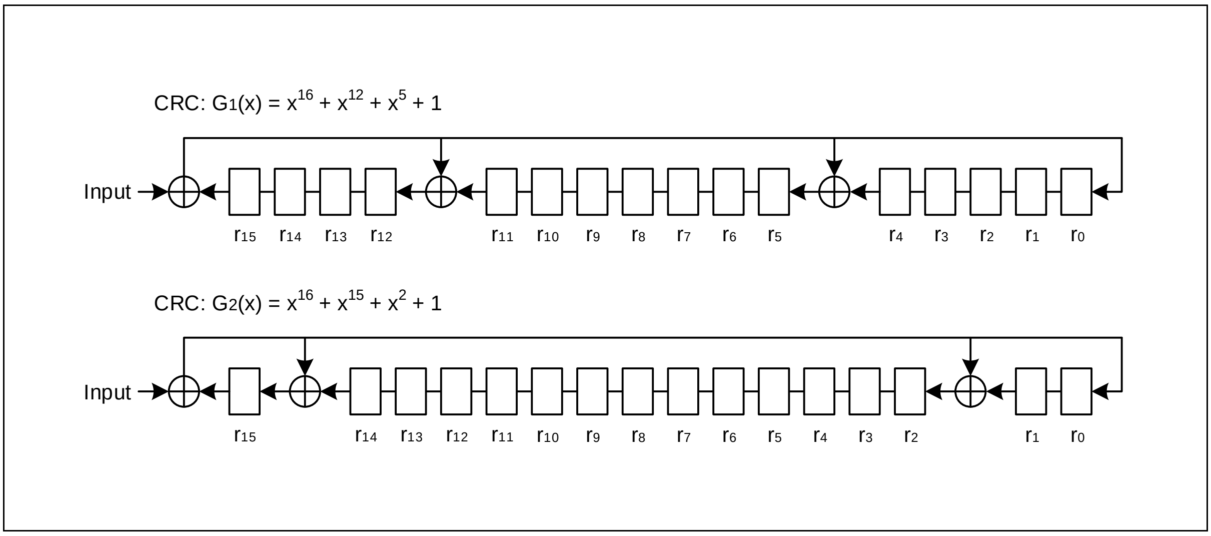
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9	8	7 6 5 4 3 2 1 0
Reserved	DMADINT	Reserved
-	0	-
-	RW	-

8	DMADINT	DMA done interrupt flag	
		0	The DMA transfer is not done.
		1	The DMA transfer is done (Writing to the bit clears the flag).

18.3 Functional description

18.3.1 CRC polynomial structure

Figure 143. CRC Polynomial Structure


CRC: G1(x) = x16 + x12 + x5 + 1
Input
r15 r14 r13 r12 r11 r10 r9 r8 r7 r6 r5 r4 r3 r2 r1 r0
CRC: G2(x) = x16 + x15 + x2 + 1
Input
r15 r14 r13 r12 r11 r10 r9 r8 r7 r6 r5 r4 r3 r2 r1 r0

18.3.2 Setting examples

<Example 1> Initial configuration of CRC

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”                : Enables SCU register access.
SCU_PER2<CRC[29]> = “1”                         : Enables the CRC peripheral.
SCU_PCER2<CRC[29]> = “1”                        : Enables the CRC peripheral clock

CRC_CTRL<POLY[2:1]> = “0”                       : Selects CRC32 operating mode.
CRC_INIT = “0x12345678”                         : Sets the CRC initial value to 0x12345678.
CRC_CTRL<INIT_EN[0]> = “1”                      : Applies the CRC initial value.
CRC_RLT = “0xABCDEF01”                          : Sets the CRC input value to 0xABCEDF01.

read_data = CRC_RLT;                            : Stores the CRC result in read_data.
```

<Example 2> DMA transfer configuration for CRC

```
SCU_SYSTEN<SYSTEN[7:0]> = “0x57”
SCU_SYSTEN<SYSTEN[7:0]> = “0x75”                : Enables SCU register access.
SCU_PER2<DMA[4]> = “1”                          : Enables the DMA peripheral.
SCU_PCER2<DMA[4]> = “1”                         : Enables the DMA peripheral clock
DMA_CR<TRANSCNT[27:16]> = “0x001”               : Sets the transfer counter number.
DMA_CR<SIZE[3:2]> = “10”                        : Sets the DMA transmit size to word size.
DMA_CR<PERISEL[12:8]> = “0x18”                  : Selects the CRC module as the peripheral for DMA transfer.
DMA_CR<DIR[1]> = “0”                            : Sets the DMA direction to memory -> peripheral (Tx).
DMA_PAR = “0x41002008(CRC_RLT)”                 : Sets the peripheral address (CRC_RLT).
DMA_MAR = “0x20000100”                          : Sets the memory address.
DMA_SR<DMAEN[0]> = “1”                          : Enables DMA.
```

19 Electrical characteristics

19.1 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 76. Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Remark
Supply voltage	VDD	-0.5 to +6	V	—
Normal pin	V_I	-0.5 to VDD+0.5	V	Voltage on any pin with respect to VSS
	V_O	-0.5 to VDD+0.5	V	
	I_{OH}	10	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	ΣI_{OH}	120	mA	Maximum current (ΣI_{OH})
	I_{OL}	20	mA	Maximum current sunk by (I_{OL} per I/O pin)
	ΣI_{OL}	120	mA	Maximum current (ΣI_{OL})
Input external main clock range	—	4 to 16	MHz	—
Storage temperature	T_{STG}	-55 to +125	℃	—
Operating temperature	Top	-40 to +105	℃	—

19.2 Recommended operating conditions

Table 77. Recommended Operating Condition

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply voltage	VDD	ADC excluded	2.5	—	5.5	V
		Only ADC	2.7	—	5.5	V
Operating frequency	FREQ	HSE	4	—	16	MHz
		HSI	31.68	32	32.32	MHz
		LSI500KHz	350	500	650	KHz
		PLL	—	—	96	MHz
Operating temperature	Top	Top	-40	—	+105	℃

19.3 ADC characteristics

Table 78. ADC Electrical Characteristics

(Temperature: -40℃ to +105℃)

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 Input buffer off	—	3.0	—	mA
		AVDD = 5.0V Input buffer on	—	4.4	—	mA
Analog input range	V_{AN}	—	VSS	—	AVDD	V
Power down current	IOFF	-	50	-		nA
Analog input capacitance	CAIN	-	14	-		pF
Conversion time	t_{CONV}	—	15*MCLK	-	45*MCLK	us
Conversion rate	F_{CONV}	AVDD > 4.0V (Input buffer on, maxf$_{AIN}$)	—	—	1.3(1.2)	MHz
		AVDD > 3.2V	—	—	1.0	MHz
		AVDD > 2.7V	—	—	0.76	MHz
Operating frequency	ACLK	—	—	—	25	MHz
DC accuracy	INL	—	—	—	±4	LSB
	DNL	—	—	—	±2	LSB
Zero offset error	ZOE	TBD	—	±4	—	LSB
Full scale error	FSE	TBD	—	±4	—	LSB

19.4 Power on Reset characteristics

Table 79. POR Electrical Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating current	I_{DD}	—	—	0.5	4	uA
POR set level	V_{set}	—	1.05	1.20	1.35	V
VDD voltage rising time	t_R	—	0.05	—	30.0	V/ms
POR reset level	V_{reset}	—	1.00	1.10	1.20	V

19.5 Low voltage reset characteristics

Table 80. Low Voltage Reset Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ.	Max	Units
Detection level	V_{LVR}	T_A = –40°C to +105°C, Falling voltage	2.01	2.12	2.23	V
			2.19	2.30	2.42	
			2.35	2.47	2.59	
			2.54	2.67	2.80	
			2.89	3.04	3.19	
			3.02	3.18	3.34	
			3.41	3.59	3.77	
			3.53	3.72	3.91	
			3.83	4.03	4.23	
			3.99	4.20	4.41	
			4.26	4.48	4.70	
Hysteresis	—	—	—	50	150	mV
Noise cancelling time	—	—	—	2	—	us
Operation current	I_{DD}	—	—	4.0	5.0	uA
Operation current(STOP)	$I_{DD, STOP}$	—	—	2.5	250	nA

19.6 Low voltage indicator characteristics

Table 81. Low Voltage Indicator Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Detection level	V_{LVI}	T_A = –40°C to +105°C, Falling voltage	2.54	2.67	2.80	V
			2.89	3.04	3.19	
			3.02	3.18	3.34	
			3.41	3.59	3.77	
			3.53	3.72	3.91	
			3.83	4.03	4.23	
			3.99	4.20	4.41	
			4.26	4.48	4.70	
Hysteresis	—		—	50	150	mV
Noise cancelling time	—		—	2	—	us
Operation current	I_{DD}		—	4.0	5.0	uA
Operation current(STOP)	$I_{DD, STOP}$		—	2.5	250	nA

19.7 Analog front end characteristics

Table 82. Op-amp Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD	—	2.5	—	5.5	V
Operating current	IDD (RMS)	AVDD=5V, 25℃	—	630	800	uA
Input offset voltage	VIO	—	—	±1.5	—	mV
Input voltage range	VI	—	0		AVDD	V
Slew rate rising	Srr	CL=200pF	—	20	—	V/us
Slew rate falling	Srf	CL=200pF	—	20	—	V/us
Gain Error	GE	Gain = x20	—	—	3	%
Common Mode Rejection Ratio	CMRR	—	—	90	—	dB
Power Supply Rejection Ratio	PSRR	—	—	90	—	dB
Gain bandwidth	fGB	CL=200pF	—	5	—	MHz
Open loop voltage gain	AV	CL=20pF	—	100	—	dB
Phase margin	tON	CL=20pF	—	80	—	Degree

Table 83. Comparator Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD	—	2.5	—	5.5	V
Input offset voltage	VIO	—	-5	—	+5	mV
Propagation delay	TDR	Rising	—	—	150	ns
	TDF	Falling	—	—	150	ns

19.8 High frequency internal RC oscillator characteristics

Table 84. High Frequency Internal RC Oscillator Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD	—	2.5	—	5.5	V
Operating current	I_{HIRC}	—	—	330	400	uA
Operating frequency	f_{32M}	-20℃ to 85℃	31.68	32	32.32	MHz
		-40℃ to 105℃	31.52	32	32.48	MHz
Frequency error	f_E	@ -20℃ to 85℃	-1.0	—	1.0	%
		@ -40℃ to 105℃	-1.5	—	1.5	%

19.9 Low frequency internal RC oscillator characteristics

Table 85. Low Frequency (500KHz) Internal RC Oscillator Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		2.5	—	5.5	V
Operating current	I_{LSI}	–	-	1.5	4.9	uA
Power down current	I_{OFFLSI}	–	-	2.6	682	nA
Operating frequency	f_{OUT}	–	350	500	650	KHz
Frequency error	f_E		-30		30	%

19.10 DC electrical characteristics

Table 86. DC Electrical Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input high voltage	V_{IH}	PA,PB,PC,PD,PF,PG,nRESET, nBOOT	0.8VDD	—	—	V
Input low voltage	V_{IL}	PA,PB,PC,PD,PF,PG,nRESET, nBOOT	—	—	0.2VDD	V
Output high voltage	V_{OH}	VDD=5V, I_{OH} = – 3mA	VDD-1.0	—	—	V
Output low voltage	V_{OL}	VDD=5V, I_{OL}=3mA	—	—	1.0	V
Output low current	I_{OL}	—	—	—	3	mA
Output high current	I_{OH}	—	-3	—	—	mA
Input high leakage current	I_{IH}	All Input ports	—	—	4	uA
Input low leakage current	I_{IL}	All Input ports	– 4	—	—	uA
Pull-up resistor	R_{PU}	R_{MAX}:V_{DD}=5.0V R_{MIN}:V_{DD}=3.0V	30	—	105	KΩ

19.11 Supply current characteristics

Table 87. Supply Current Characteristics: Normal and Sleep Mode

Parameter	Symbol	Condition	HCLK	Typ.	Max	unit
Normal operation	IDD_{RUN}	External clock with PLL code running from Flash Low-speed(LSI500K) and High-speed(HSI32M) internal oscillator OFF PCLK = HCLK	96MHz	15.4	30.8	mA
			60MHz	10.4	20.8	
			30MHz	6.2	12.4	
			16MHz	4.2	8.4	
		External clock(PLL not used) code running from Flash Low-speed(LSI500K) and High-Speed(HSI32M) internal oscillator OFF PCLK = HCLK	8MHz	2.9	5.8	
		High-speed(HSI32M) internal oscillator clock code running from Flash Low-speed(LSI500K) internal oscillator and External clock OFF PCLK = HCLK	32MHz	5.8	11.6	
		Low-speed(LSI500K) internal oscillator clock code running from Flash High-speed internal(HSI32M) oscillator and External clock OFF PCLK = HCLK	500KHz	324	648	uA
Sleep mode	IDD_{SLEEP}	External clock with PLL code running from Flash Low-speed(LSI500K) and High-speed(HSI32M) internal oscillator OFF PCLK = HCLK	96MHz	8.6	17.2	mA
			60MHz	6.1	12.2	
			30MHz	4	8	
			16MHz	3	6	
		External clock(PLL not used) code running from Flash Low-speed(LSI500K) and High-Speed(HSI32M) internal oscillator OFF PCLK = HCLK	8MHz	2.1	4.2	
		High-speed(HSI32M) internal oscillator clock code running from Flash Low-speed(LSI500K) internal oscillator and External clock OFF PCLK = HCLK	32MHz	2.5	5	

Table 87. Supply Current Characteristics: Normal and Sleep Mode (continued)

Parameter	Symbol	Condition	HCLK	Typ.	Max	unit
Sleep mode (cont.)	IDD_{SLEEP} (cont.)	Low-speed(LSI500K) internal oscillator clock code running from Flash High-speed internal(HSI32M) oscillator and External clock OFF PCLK = HCLK	500KHz	272	544	uA

NOTES:

1. Typical values are measured at T_A=25℃, and V_{DD}=5V
2. All I/O pins are in output mode and output low status.
3. All peripherals are disabled.
4. The Flash access time is adjusted to HCLK frequency (0 wait state from 0 to 28MHz, 1 wait state from 28 to 56MHz and 2 wait states above)

Table 88. Supply Current Characteristics: Stop Mode

Parameter	Symbol	Condition	Typ.	Max		unit
				TA=25℃	TA=105℃	
Stop mode	IDD_{STOP}	Internal VDC regulator disable mode, Low-speed and high-speed internal RC oscillators and External clock OFF (no independent watchdog and Low voltage reset)	100	350	3000	uA

NOTES:

1. Typical values are measured at T_A=25℃, and V_{DD}=5V
2. All I/O pins are in output mode and output low status.
3. All peripherals are disabled.
4. The Flash access time is adjusted to HCLK frequency (0 wait state from 0 to 28MHz, 1 wait state from 28 to 56MHz and 2 wait states above)

19.12 Internal Flash ROM characteristics

Table 89. Internal Flash ROM Characteristics

(Temperature: -40℃ to +105℃)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Max available clock frequency	—	0-wait	—	—	28	MHz
Reset cycle time	$t_{RSTBUSY}$	—	8	—	—	us
Fuse program time	$t_{FRDBUSY}$		—	—	6	us
Normal program time	$t_{PGMBUSY}$	—	—	—	25	us
Burst program time	$t_{BMPGMBUSY}$		—	—	15	us
Normal page erase time	$t_{PERSBUSY}$	—	—	—	2	ms
Sector erase time	$t_{SERSBUSY}$	—	—	—	2	ms
Mat erase time	$t_{MERSBUSY}$	—	—	—	8	ms
Endurance of write/erase	N_{FWE}	T_A=25 °C, Page unit	10,000	—	—	Times
Retention time	t_{FRT}	—	10	—	—	Years

19.13 Main oscillator characteristics

Table 90. Main Oscillator Characteristics

(Temperature: -40℃ to +105℃)

Oscillator	Parameter	Conditions	Min	Typ.	Max	Units
Operating voltage	VDD	—	1.8	5.0	5.5	V
Operating current	IDD	VDD≥2.4V **NOTE**		0.91	1.5	mA
Power down current	I_{STOP}	—	—	0.2	30	nA
Output frequency	XOUT input frequency	VDD≥2.4V **NOTE**	1.0	—	16.0	MHz
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	4M≤fOUT≤12M	10	22	30	pF
		12M≤fOUT≤16M	7	18	22	pF
Feedback resistance	R_{FB}	VDD=5V	0.7	1.0	1.3	㏁

NOTE: EISEL = 0x0, ENFSEL = 0x3

Figure 144. Crystal/Ceramic Oscillator

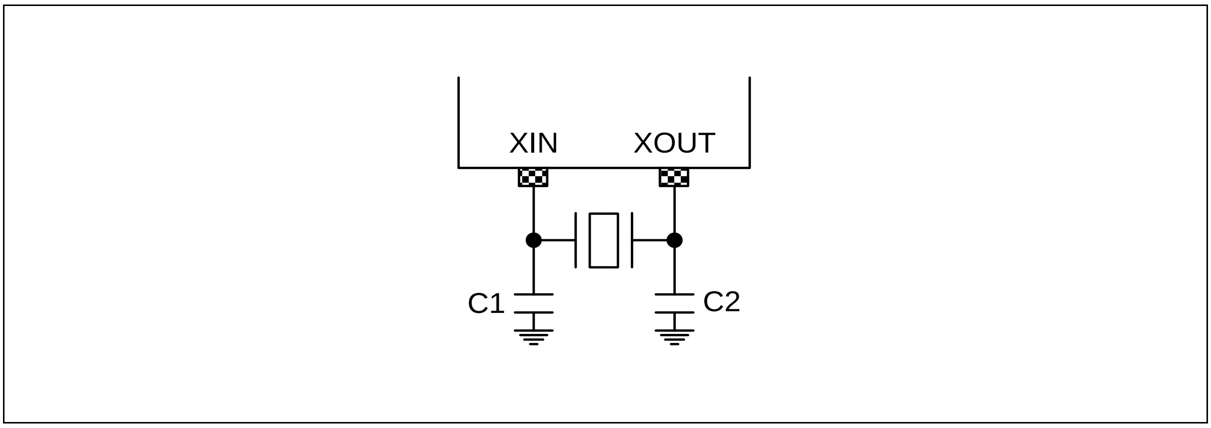


Figure 145. External Clock

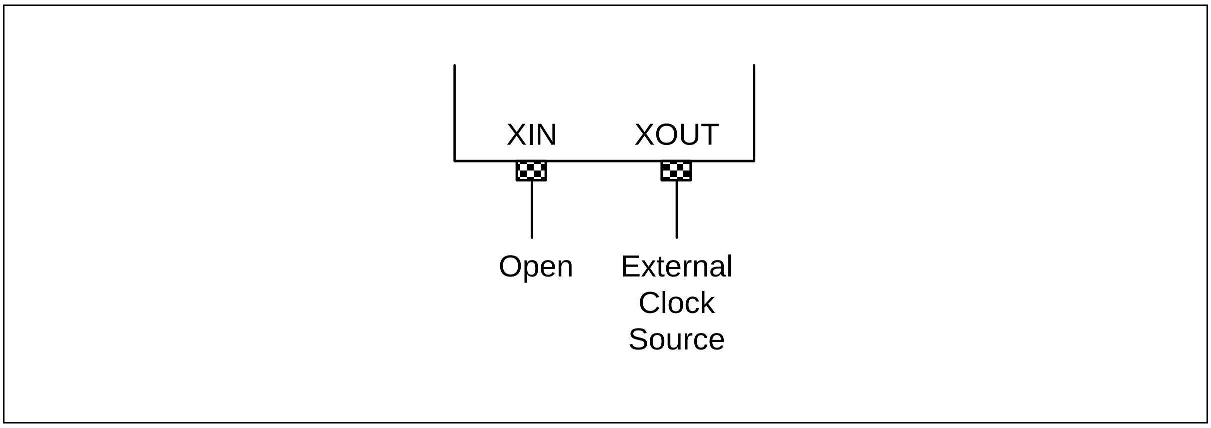
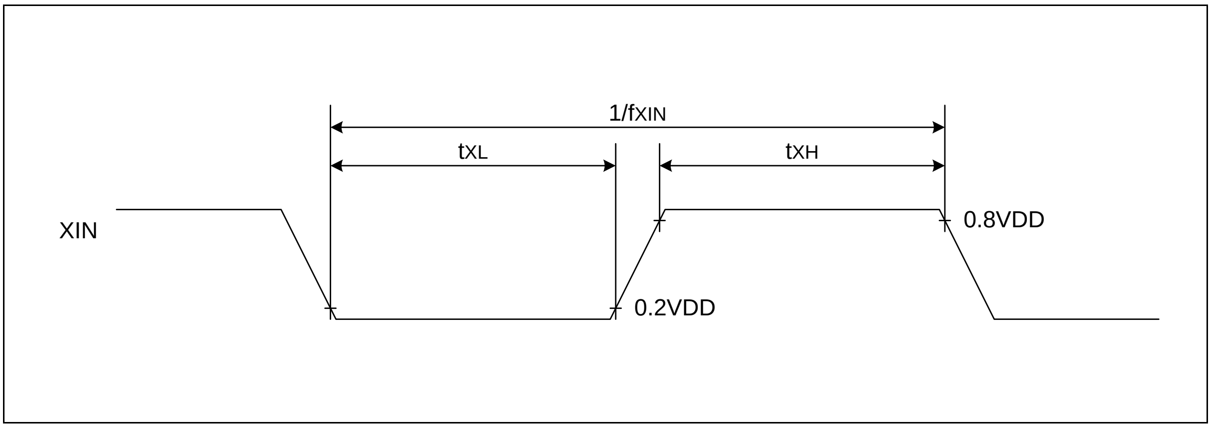


Figure 146. Clock Timing Measurement at XIN



19.14 PLL electrical characteristics

Table 91. PLL Electrical Characteristics

(Temperature: -40℃ to +105℃)

Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Units**
Operating voltage	VDD	—	2.5	—	5.5	V
Operating current	I_{DD}	—	—	0.5	1	mA
Output frequency	f_{OUT}	—		—	96	MHz
Duty	f_{DUTY}	—	40	—	60	%
Input frequency	$f_{PLLINCLK}$	—	4	8	16	MHz
FXIN frequency	f_{XIN}	—	1	2	3	MHz
P-P jitter	t_{JITTER}	@Lock State	—	—	500	ps

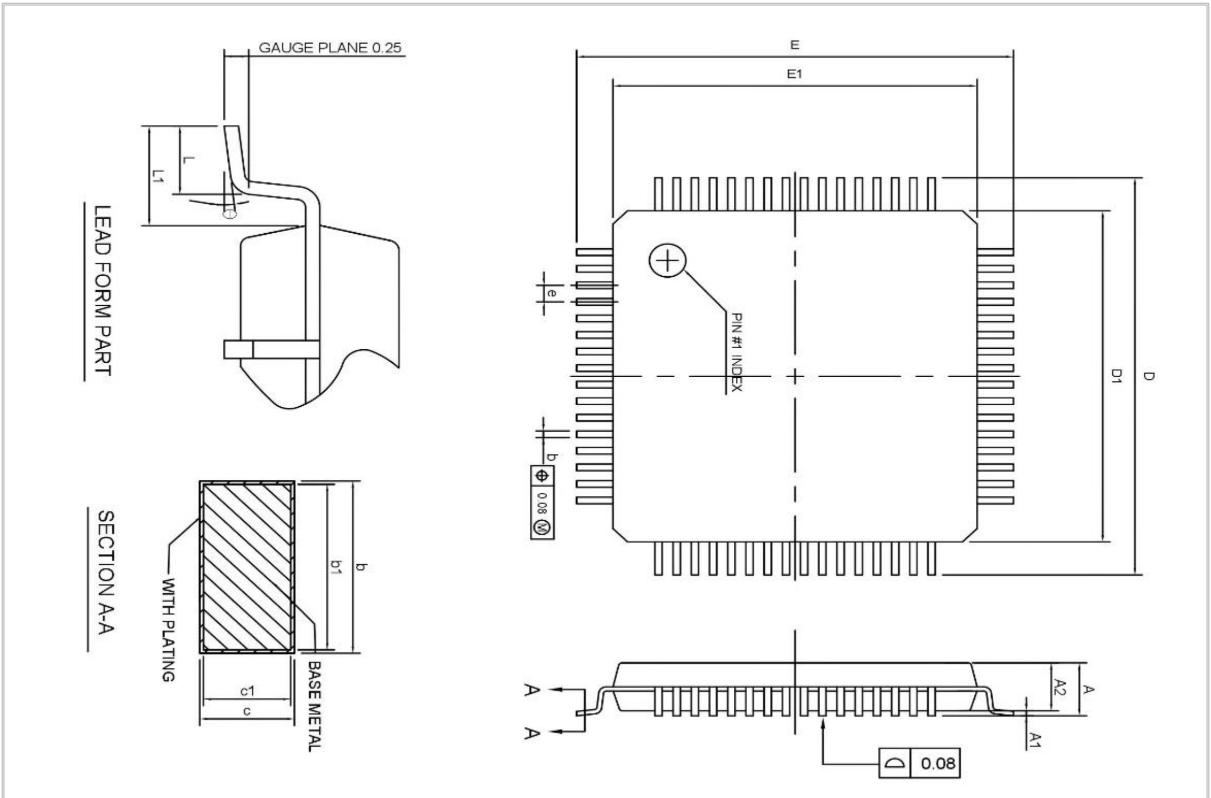
NOTE: f_{XIN} = $f_{PLLINCLK}$/Pre divider value

20 Package information

20.1 64 LQFP package information

Figure 147. 64 LQFP (10x10) Package Outline

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE	04/18/'17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	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°

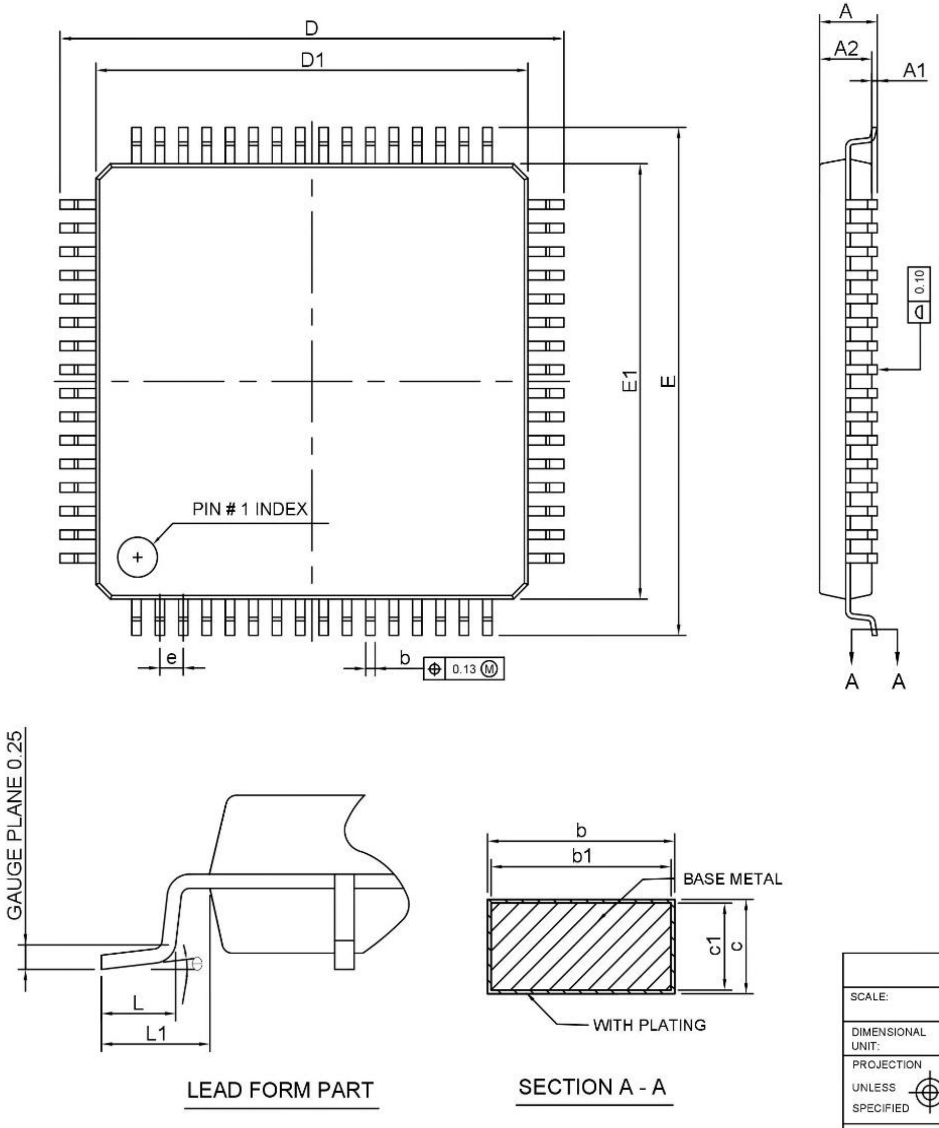
NOTES:

1. ALL DIMENSION REFER TO JEDEC STANDARD MS-026-BCD.
2. DIMENSIONS 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25MM 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 0.08MM.
4. 'A1' IS DEFINED AS THE DISTANCE FROMTHE SEATING PLANE TO THE LOWEST POINT ON THE PACKAGE BODY.

LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/18/'17	DRAWN: JS. KANG	TITLE: LQFP 10X10MM 64LD PACKAGE OUTLINE	
DIMENSIONAL UNIT: MM	UN.TOLERANCED DIMENSIONS	ENGINEER: JH. KIM		
PROJECTION	FRAC:	CHECKED: YJ. CHAE		
UNLESS SPECIFIED	.XX .XXX .XXXX ANGLE:	APPROVED: YJ. CHAE		
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1010064	DRAWING NUMBER: OLQ-1010064	REV: — SHEET: 1 OF 1

Figure 148. 64 LQFP (12x12) Package Outline



REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	04/18/'17	JH. KIM	YJ. CHAE

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.22	0.32	0.38
b1	0.25	0.30	0.33
c	0.09	—	0.20
c1	0.09	—	0.16
D	13.80	14.00	14.20
D1	11.80	12.00	12.20
E	13.80	14.00	14.20
E1	11.80	12.00	12.20
e	0.65 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°

NOTES:

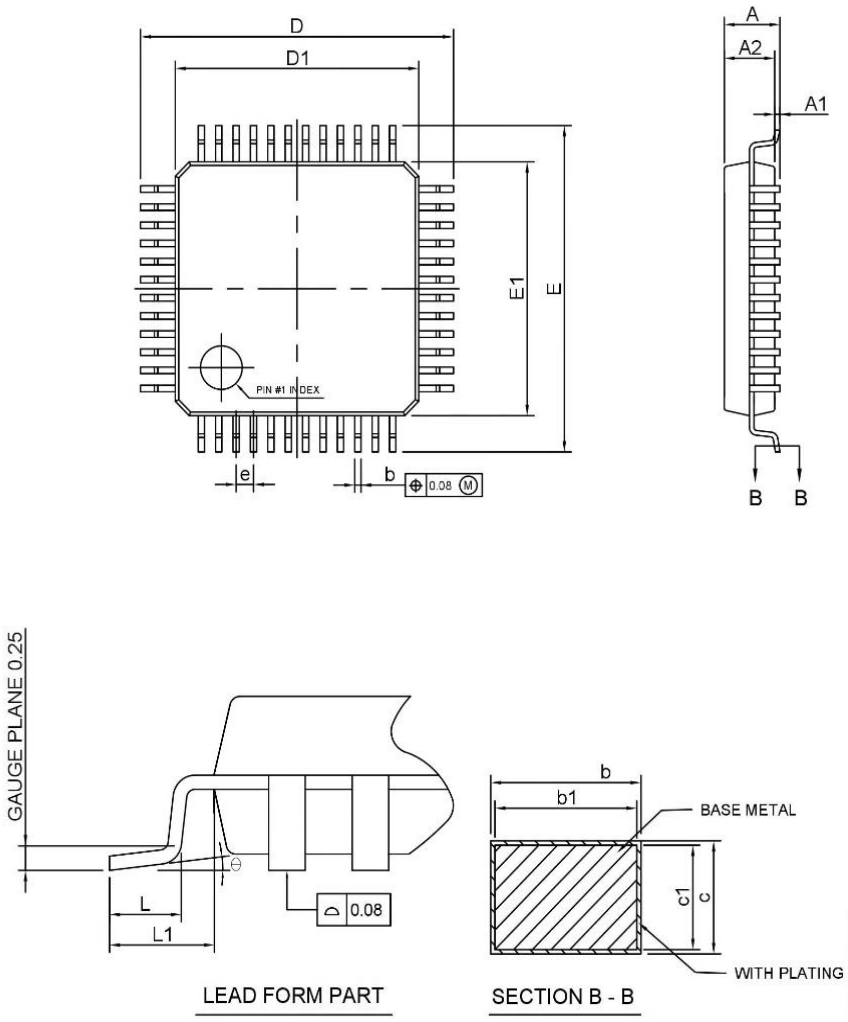
1. ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026-BDC.
2. DIMENSIONS 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD FLASH.
3. DIMENSION 'b' DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM 'b' DIMENSION BY MORE THAN 0.08MM.
4. 'A1' IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE TO THE LOWEST POINT ON THE PACKAGE BODY.

LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/18/'17	DRAWN: JS. KANG	TITLE: LQFP 12X12MM 64LD PACKAGE OUTLINE		
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER: JH. KIM			
PROJECTION UNLESS SPECIFIED	FRAC: .XX .XXX .XXXX ANGLE	CHECKED: YJ. CHAE APPROVED: YJ. CHAE			
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1212064	DRAWING NUMBER: OLQ-1212064	REV. —	SHEET: 1 OF 1

20.2 48 LQFP package information

Figure 149. 48 LQFP (07x07) Package Outline



REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	04/18/'17	JH. KIM	YJ. CHAE

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	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 DIMENSIONS REFER TO JEDEC STANDARD MS-026-BBC.
2. DIMENSION 'D1' AND 'E1' DO NOT INCLDUE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION 'b' DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMABR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM 'b' DIMENSION BY MORE THAN 0.08MM.
4. 'A1' IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT ON THE PACKAGE BODY.

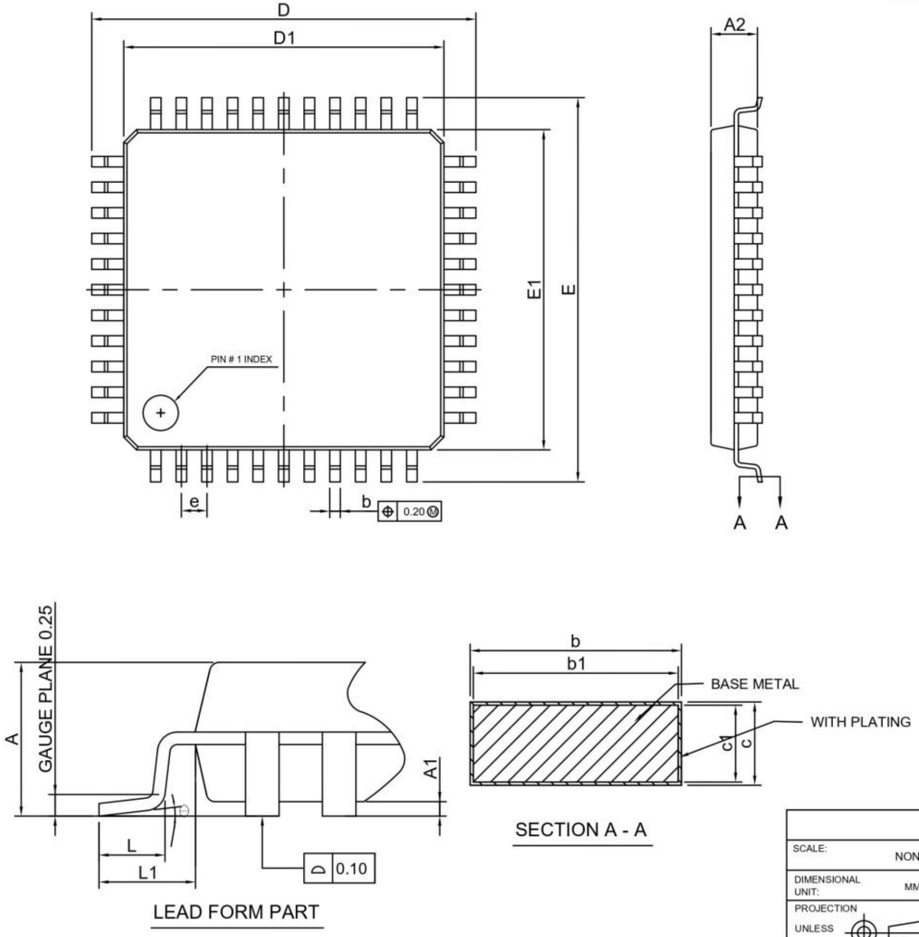
LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE:	DRAWN: JS. KANG	TITLE: LQFP 7X7MM 48LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER: YJ. CHAE	
PROJECTION UNLESS SPECIFIED	FRAC: .XX .XXX .XXXX ANGLE	CHECKED: YJ. CHAE	
		APPROVED: JH. KIM	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-0707048	DRAWING NUMBER: OLQ-0707048 / REV. — / SHEET: 1 OF 1

20.3 44 LQFP package information

Figure 150. 44 LQFP (10x10) Package Outline

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	04/18/'17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.22	0.37	0.45
b1	0.22	0.35	0.40
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		0.80 BSC	
L	0.45	0.60	0.75
L1		1.00 REF	
θ	0°	3.5°	8°

NOTES:

1. ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026-BCB.
2. DIMENSIONS 'D1' AND 'E1' DON NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM 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 0.08MM.
4. 'A1' IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT ON THE PAKCAGE BODY.

LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/18/'17	DRAWN: JS. KANG	TITLE: LQFP 10X10MM 44LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED	FRAC: .XX .XXX .XXXX ANGLE	CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1010044	DRAWING NUMBER: OLQ-1010044 REV. — SHEET: 1 OF 1

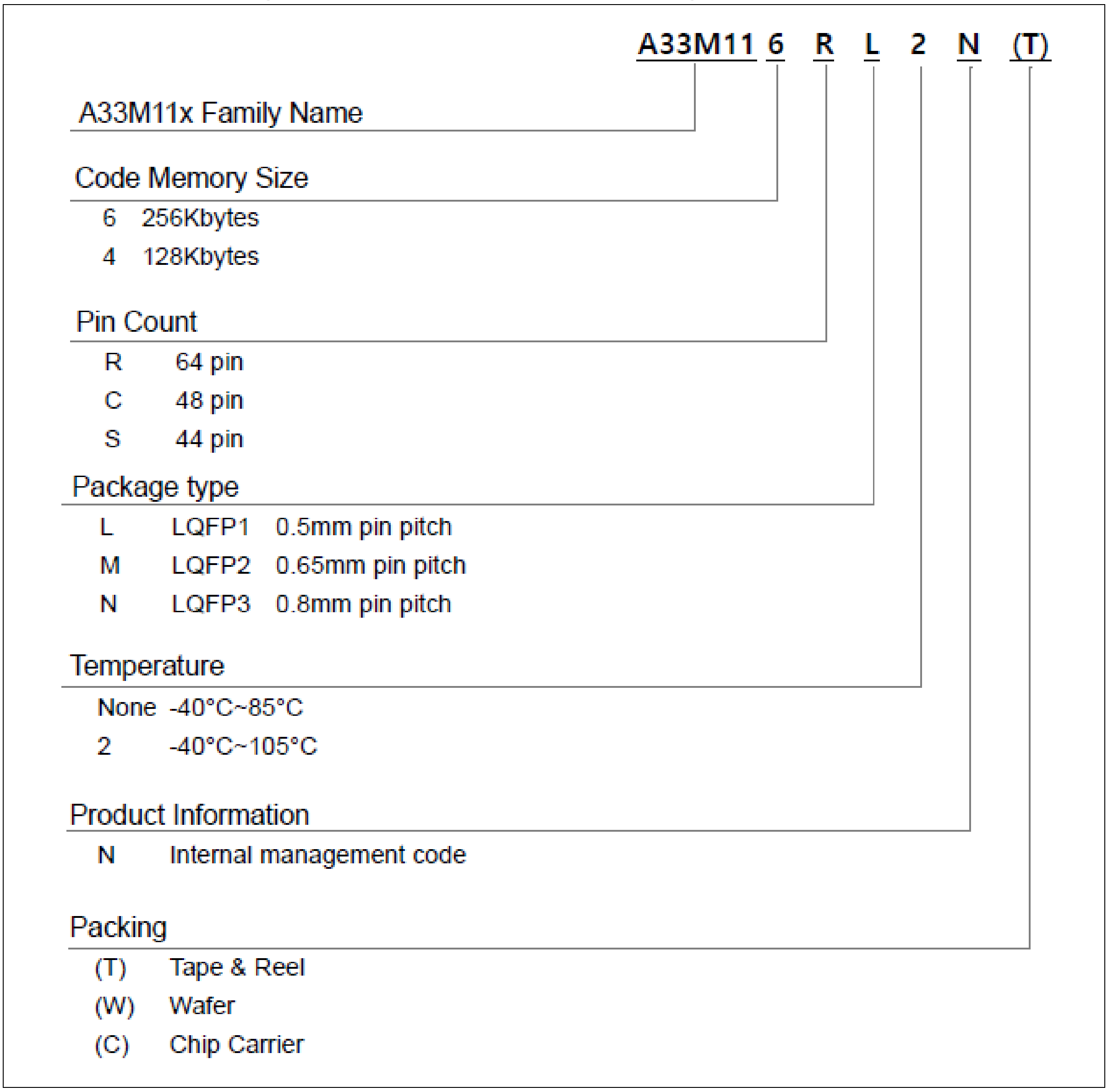
21 Ordering information

Table 92. A33M11x Series Device Ordering Information

Device name	Flash	SRAM	SPI	UART	I2C	OPAMP (COMP)	TIMER	MPWM	ADC	I/O ports	Package
A33M116RL	256KB	16KB	2	4	2	4	8	2	16	56	LQFP-64
A33M116RM*	256KB	16KB	2	4	2	4	8	2	16	56	LQFP-64
A33M114RL*	128KB	16KB	2	4	2	4	8	2	16	56	LQFP-64
A33M116CL*	256KB	16KB	2	3	2	4	8	2	20	45	LQFP-48
A33M114CL*	128KB	16KB	2	3	2	4	8	2	20	45	LQFP-48
A33M114SN*	128KB	16KB	2	3	2	4	8	2	16	41	LQFP-44

* : For available options or further information on the devices marked with "*", please contact the ABOV sales offices.

Figure 151. A33M11x Device Numbering Nomenclature



22 Development tools

This chapter introduces wide range of development tools for A33M11x. 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.

22.1 Compiler

ABOV semiconductor does not provide any compiler for A33M11x. However, since A33M11x have ARM's high-speed 32-bit Cortex-M3 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.

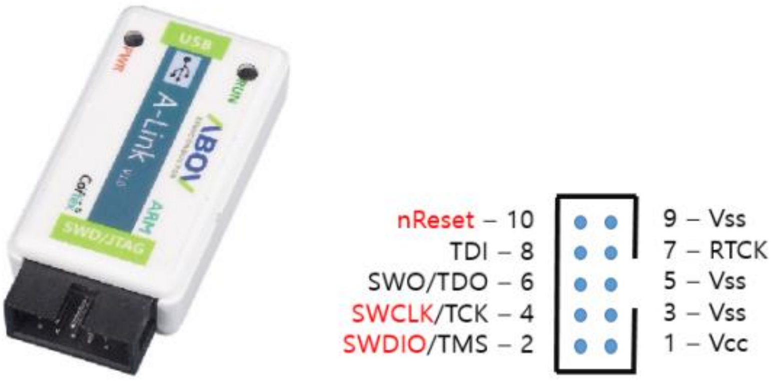
22.2 Debugger

The A-Link and A-Link Pro support ABOV Semiconductor's A33M11x 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 152. 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 152. A-Link and Pin Descriptions



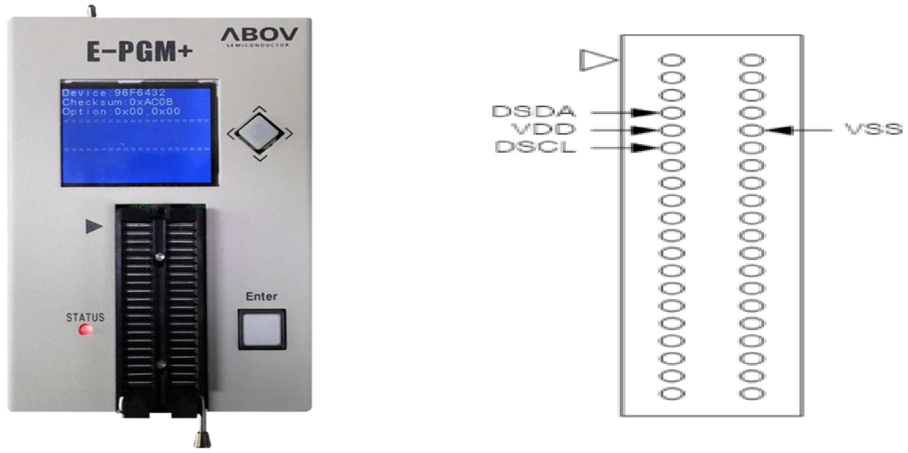
22.3 Programmer

22.3.1 E-PGM+

E-PGM+ is a single programmer, and allows a user to program on the device directly.

- Support ABOV devices
- 2 to 5 times faster than S-PGM+
- Main controller: 32-bit MCU @ 72MHz
- Buffer memory: 1MB

Figure 153. E-PGM+ (Single Writer) and Pin Descriptions



22.3.2 Gang programmer

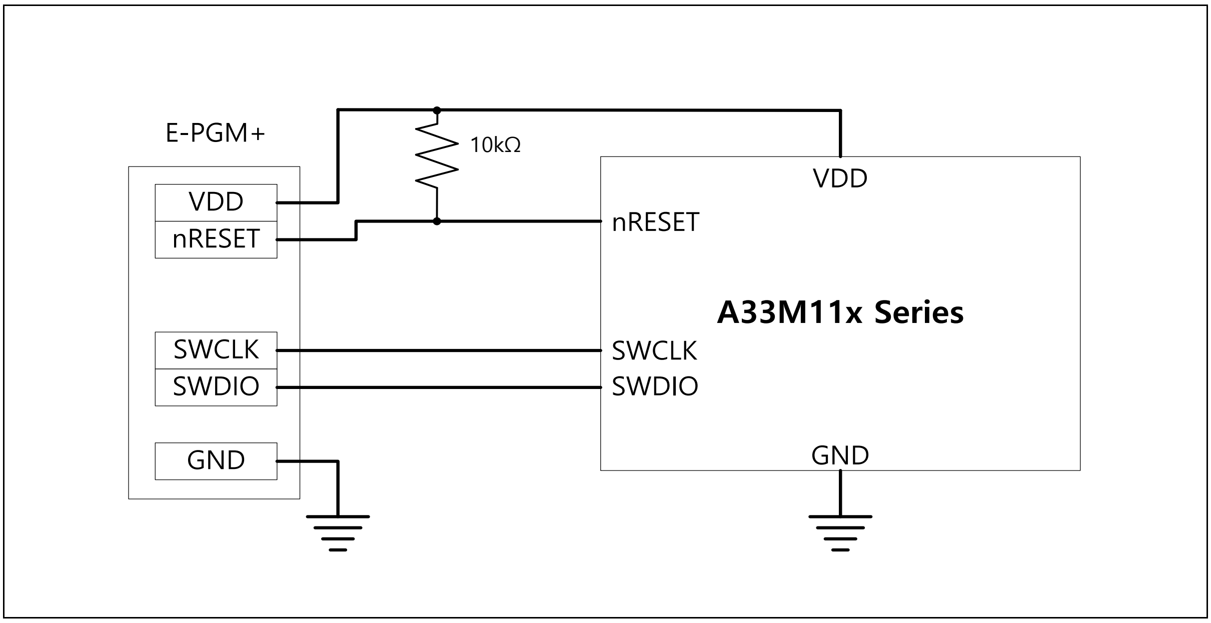
E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.

Figure 154. E-Gang4 and E-Gang6: for Mass Production

22.4 SWD debug mode and E-PGM+ connection

Connections for SWD debugger interface or E-PGM+ is described in Figure 155.

Figure 155. Connection between A33M11x Series and E-PGM+ using SWD Debugger Interface



Revision history

Version	Date	Description
1.00	Jun. 25, 2020	1st creation
1.01	Dec. 15, 2020	4.5.19 Clock output divider value updated 4.5.22 PLL Setting sequence updated 5.3.1,5.3.2 Port mux description added 5.3.5 Port debounce description added 5.3.5 Port access enable description added 5.3.7 Port Pending/non-pending description added Table27 Register type updated 6.1.6,6.2.4 Bit Type updated 8.3.6,9.3.2 Typing error updated 14.2.18 Typing error updated
1.02	May. 5, 2021	6.1.1 Cache clear updated 16.3.3 ADC precautions updated Table 70 ADC trigger updated Table 2 Pin description note updated 4.2.4 Clock setting guide updated Table 24, Table 25 Code Flash and data Flash PGM unit updated 8.3.6 Prescaler clock source updated 10.2.2 Frt match counter description updated 11.2.2 Uart THR bit name updated Table 47. Interrupt IDs and ControlUart interrupt IDs and control updated 11.2.8 Uart baud rate divisor updated 11.3.1,11.3.3,11.3.4,11.3.5,11.3.6 Uart functional description Updated 4.5.17,5.3.1,5.3.2,5.3.3,17.2.1 Port notes added
1.03	Dec. 2, 2021	Figure 3, Figure 4 Note added 2.2 Pin description updated 17.2 Afe reset value updated Table 10. warm reset updated 6.1.4 first 4KB typo updated 6.2.1 WRITE bit added 12.2.5 SPI note added 15.2.1 INXGATE typo updated
1.04	Jan. 13, 2023	Changed style and font
1.05	Mar. 28, 2023	Added 44 LQFP package information Updated typo error 5.3.1,5.3.2

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 disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of ABOV products in such unauthorized applications. ABOV and the ABOV logo are trademarks of ABOV. 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.