



CMS32M65xx User Manual

Ultra-low power 32-bit microcontrollers based on ARM® Cortex®-M0+
V1.0.1

Please note the following CMS IP policy

* China Micro Semicon Co., Ltd. (hereinafter referred to as the Company) has applied for patents and holds absolute legal rights and interests. The patent rights associated with the Company's MCUs or other products have not been authorized for use, and any company, organization, or individual who infringes the Company's patent rights through improper means will be subject to all possible legal actions taken by the Company to curb the infringement and to recover any damages suffered by the Company as a result of the infringement or any illegal benefits obtained by the infringer.

* The name and logo of Cmsemicon are registered trademarks of the Company.

* The Company reserves the right to further explain the reliability, functionality and design improvements of the products in the data sheet. However, the Company is not responsible for the use of the Specification Contents. The applications mentioned herein are for illustrative purposes only and the Company does not warrant and does not represent that these applications can be applied without further modification, nor does it recommend that its products be used in places that may cause harm to persons due to malfunction or other reasons. The Company's products are not authorized for use as critical components in lifesaving, life-sustaining devices or systems. The Company reserves the right to modify the products without prior notice. For the latest information, please visit the official website at www.mcu.com.cn.

Documentation Instructions

This manual is a technical reference manual for the CMS32M65xx microcontroller product. The technical reference manual is the application instruction material on how to use this series of products, including the structure, function description, working mode and register configuration of each functional module.

The technical reference manual is a description of all functional modules of this series of products. If you want to know the feature description of the product (that is, the functional configuration), you can refer to the respective data sheet.

The data sheet information is as follows:

CMS32M65xx: CMS32M65xx_datasheet_vx.x.x. pdf

Usually in the early stage of chip selection, you shall first check the data sheet to evaluate whether the product can meet the functional requirements of the design; after basically selecting the required product, you need to check the technical reference manual to determine whether the working mode of each functional module does meet the requirement; When determining the selection and entering the programming design stage, you need to read the technical reference manual in detail to understand the specific implementation and register configuration of each function. Refer to the data sheet for information on voltages, currents, drive capabilities, and pin assignments when designing hardware.

For a detailed description of the Cortex-M0+ core, SysTick timer and NVIC, please refer to the respective ARM documents.

Table of Contents

| | |
|-------------------------------------------------------------------------------|-----------|
| Documentation Instructions | 2 |
| Chapter 1 CPU | 14 |
| 1.1 Overview | 14 |
| 1.2 Cortex-M0+ core features | 14 |
| Chapter 2 Debugging Features | 15 |
| 2.1 SWD interface pins | 16 |
| 2.2 ARM reference documents | 17 |
| Chapter 3 Pin Function | 18 |
| 3.1 Port function | 18 |
| 3.2 Port multiplexing function | 18 |
| 3.3 Register mapping | 20 |
| 3.3.1 Control function register mapping | 20 |
| 3.3.2 Output-input multiplexing function register mapping | 22 |
| 3.3.3 Special function port RESINB control register mapping | 24 |
| 3.4 Register description | 25 |
| 3.4.1 Port register (Px) | 26 |
| 3.4.2 Port mode register (PMx) | 27 |
| 3.4.3 Pull-up resistor selection register (PUx) | 28 |
| 3.4.4 Pull-down resistor selection register (PDx) | 29 |
| 3.4.5 Port output mode register (POMx) | 30 |
| 3.4.6 Port mode control register (PMCx) | 31 |
| 3.4.7 Port set control register (PSETx) | 32 |
| 3.4.8 Port clear control register (PCLR _x) | 33 |
| 3.4.9 Port output multiplexing configuration register (PmnCFG) | 34 |
| 3.4.10 Port input multiplexing configuration register (PSxx_CFG) | 35 |
| 3.4.11 TTL and Schmitt input selection (PxTTLCFG,x=0 and 2) | 39 |
| 3.4.12 Special function port RESINB description (RSTM) | 39 |
| Chapter 4 System Architecture | 40 |
| 4.1 Overview | 40 |
| 4.2 System address partitioning | 41 |
| Chapter 5 Clock Generation Circuit | 44 |
| 5.1 Function of clock generation circuit | 44 |
| 5.2 Configuration of clock generation circuit | 45 |
| 5.3 Register mapping | 47 |
| 5.4 Register description | 48 |
| 5.4.1 System clock control register (CKC) | 48 |
| 5.4.2 Clock operation status control register (CSC) | 49 |
| 5.4.3 Peripheral enable registers 0, 1 (PER0, PER11, PER12, PER13) | 50 |
| 5.4.4 12-bit interval timer operation clock select register (OSMC) | 54 |
| 5.4.5 High-speed on-chip oscillator frequency select register (HOCODIV) | 55 |
| 5.4.6 Low-speed on-chip oscillator clock select register (SUBCKSEL) | 56 |
| 5.4.7 Power mode control protection register (PMUKEY) | 56 |
| 5.4.8 Power mode control register (PMUCTL) | 56 |

| | | |
|------------------|----------------------------------------------------------------------------|-----------|
| 5.5 | System clock oscillation circuit..... | 57 |
| 5.5.1 | High-speed on-chip oscillator..... | 57 |
| 5.5.2 | Low-speed on-chip oscillator..... | 57 |
| 5.6 | Operation of clock generation circuit..... | 58 |
| 5.7 | Clock control..... | 59 |
| 5.7.1 | Example of setting up a high-speed on-chip oscillator | 59 |
| 5.7.2 | CPU clock status transition diagram | 61 |
| 5.7.3 | Conditions before CPU clock transfer and post-transfer processing..... | 64 |
| 5.7.4 | Time required to switch CPU clock and main system clock | 65 |
| 5.7.5 | Conditions before clock oscillation is stopped | 65 |
| Chapter 6 | General-Purpose Timer Unit | 66 |
| 6.1 | Function of general-purpose timer unit | 66 |
| 6.1.1 | Independent channel operation function | 66 |
| 6.1.2 | Multi-channel linkage operation functions | 68 |
| 6.1.3 | 8-bit timer operation function (channels 1 and 3 of unit 0 only)..... | 70 |
| 6.2 | Structure of general-purpose timer unit | 71 |
| 6.2.1 | Register mapping | 73 |
| 6.2.2 | Timer count register mn (TCRmn) | 74 |
| 6.2.3 | Timer data register mn (TDRmn) | 75 |
| 6.2.4 | Peripheral enable register 0 (PER0) | 76 |
| 6.2.5 | Timer clock select register m (TPSm) | 77 |
| 6.2.6 | Timer mode register mn (TMRmn)..... | 79 |
| 6.2.7 | Timer status register mn (TSRmn)..... | 91 |
| 6.2.8 | Timer channel enable status register m (TEm)..... | 92 |
| 6.2.9 | Timer channel start register m (TSm) | 93 |
| 6.2.10 | Timer channel stop register m (TTm)..... | 95 |
| 6.2.11 | Timer input/output output select register (TIOS0)..... | 96 |
| 6.2.12 | Timer output enable register m (TOEm) | 97 |
| 6.2.13 | Timer output register m (TOm)..... | 98 |
| 6.2.14 | Timer output level register m (TOLm) | 99 |
| 6.2.15 | Timer output mode register m (TOMm) | 100 |
| 6.2.16 | Noise filter enable register 1 (NFEN1) | 101 |
| 6.2.17 | Registers controlling port functions of timer input/output pins | 102 |
| 6.3 | Basic rules of general-purpose timer unit | 103 |
| 6.3.1 | Basic rules of multi-channel linkage operation function..... | 103 |
| 6.3.2 | Basic rules of 8-bit timer operation function (channels 1 and 3 only)..... | 105 |
| 6.4 | Operation of counter | 106 |
| 6.4.1 | Count clock (F_{TCLK}) | 106 |
| 6.4.2 | Start timing of counter | 108 |
| 6.4.3 | Operation of counter..... | 109 |
| 6.5 | Channel output (TOmn pin) control..... | 115 |
| 6.5.1 | TOmn pin output circuit configuration | 115 |
| 6.5.2 | TOmn pin output setting | 116 |
| 6.5.3 | Cautions on channel output operation | 117 |
| 6.5.4 | One-time operation of TOmn bit..... | 122 |
| 6.5.5 | Timer interrupt and TOmn pin output when counting starts | 123 |
| 6.6 | Control of timer input (TImn) | 124 |

| | | |
|------------------|------------------------------------------------------------------------------------------|------------|
| 6.6.1 | Structure of TlMn pin input circuit | 124 |
| 6.6.2 | Noise filter..... | 124 |
| 6.6.3 | Cautions on channel input operation..... | 125 |
| 6.7 | Independent channel operation function of general-purpose timer unit | 126 |
| 6.7.1 | Operation as interval timer/square wave output | 126 |
| 6.7.2 | Operation as external event counter | 130 |
| 6.7.3 | Operation as frequency divider | 133 |
| 6.7.4 | Operation as input pulse interval measurement | 136 |
| 6.7.5 | Operation as input signal high-/low-level width measurement | 140 |
| 6.7.6 | Operation as delay counter | 144 |
| 6.8 | Multi-channel linkage operation function for general purpose timer unit..... | 147 |
| 6.8.1 | Operation as single trigger pulse output function..... | 147 |
| 6.8.2 | Operation as PWM function | 154 |
| 6.8.3 | Operation as multiple PWM output function..... | 162 |
| Chapter 7 | LSITIMER 12-Bit Interval Timer | 170 |
| 7.1 | Function of 12-bit interval timer..... | 170 |
| 7.2 | Configuration of 12-bit interval timer | 170 |
| 7.3 | Register mapping | 170 |
| 7.4 | 12-bit interval timer control register (CON0) | 171 |
| 7.5 | 12-bit interval timer operation | 172 |
| 7.5.1 | 12-bit interval timer operation timing | 172 |
| 7.5.2 | Start of count operation and re-enter to sleep mode after returned from sleep mode | 173 |
| Chapter 8 | Clock Output/Buzzer Output Controller | 174 |
| 8.1 | Function of clock output/buzzer output controller | 174 |
| 8.2 | Structure of clock output/buzzer output controller | 175 |
| 8.3 | Register mapping | 176 |
| 8.3.1 | Clock output select register (CKS0) | 176 |
| 8.3.2 | Clock output select register (CKS1) | 177 |
| 8.4 | Registers for configuring clock output/buzzer output port functions..... | 178 |
| 8.5 | Operation of clock output/buzzer output controller | 179 |
| 8.5.1 | Operation as output pin | 179 |
| 8.6 | Cautions of clock output/buzzer output controller..... | 179 |
| Chapter 9 | Watchdog Timer | 180 |
| 9.1 | Function of watchdog timer | 180 |
| 9.2 | Configuration of watchdog timer | 180 |
| 9.3 | Register mapping | 182 |
| 9.3.1 | Watchdog timer enable register (WDTE) | 182 |
| 9.3.2 | LOCKUP control register (LOCKCTL)..... | 183 |
| 9.3.3 | Protection register (PRCR)..... | 183 |
| 9.3.4 | Watchdog configuration register (WDTCFG0/1/2/3) | 184 |
| 9.4 | Operation of watchdog timer | 185 |
| 9.4.1 | Operational control of watchdog timer | 185 |
| 9.4.2 | Setting overflow time of watchdog timer | 186 |
| 9.4.3 | Setting window open period of watchdog timer | 187 |
| 9.4.4 | Setting watchdog timer interval interrupt..... | 188 |
| 9.4.5 | Operation of watchdog timer during LOCKUP | 188 |

| | |
|---------------------------------------------------------|------------|
| Chapter 10 DIVSQRT Unit | 189 |
| 10.1 Overview | 189 |
| 10.2 Features | 189 |
| 10.3 Functional description | 189 |
| 10.4 Register mapping | 190 |
| 10.5 Register description | 191 |
| 10.5.1 DIVSQRT control register (DIVSQRCON) | 191 |
| 10.5.2 DIVSQRT data A register (DIVSQRALUA) | 191 |
| 10.5.3 DIVSQRT data B register (DIVSQRALUB) | 191 |
| 10.5.4 DIVSQRT result 0 register (DIVSQRTRES0) | 191 |
| 10.5.5 DIVSQRT result 1 register (DIVSQRTRES1) | 191 |
| Chapter 11 Timer (TIMER0/1) | 192 |
| 11.1 Overview | 192 |
| 11.2 Features | 192 |
| 11.3 Function description | 192 |
| 11.3.1 Single trigger mode | 192 |
| 11.3.2 Periodic count mode | 192 |
| 11.3.3 Continuous count mode | 192 |
| 11.3.4 Delayed load function | 193 |
| 11.4 Register mapping | 194 |
| 11.5 Register description | 195 |
| 11.5.1 Timer control register (CON0/1) | 195 |
| 11.5.2 Timer load register (LOAD0/1) | 195 |
| 11.5.3 Timer current value register (VAL0/1) | 195 |
| 11.5.4 Timer interrupt source status register (RIS0/1) | 196 |
| 11.5.5 Timer enabled interrupt status register (MIS0/1) | 196 |
| 11.5.6 Timer interrupt clear register (ICLR0/1) | 196 |
| 11.5.7 Timer delayed load register (BGLOAD0/1) | 196 |
| Chapter 12 Capture/Compare/PWM Module (CCP0/1) | 197 |
| 12.1 Overview | 197 |
| 12.2 Features | 197 |
| 12.3 Function description | 198 |
| 12.3.1 Pulse Width Modulation (PWM) | 198 |
| 12.3.2 Square wave output mode | 198 |
| 12.3.3 Capture mode 0 | 199 |
| 12.3.4 Capture mode 1 | 199 |
| 12.3.5 Capture mode 2 | 201 |
| 12.3.6 PWM configuration process | 203 |
| 12.3.7 Interrupt | 203 |
| 12.4 Register mapping | 204 |
| 12.5 Register description | 205 |
| 12.5.1 CCPx control register (CCPCONx) (x=0,1) | 205 |
| 12.5.2 CCP reload register (CCPLOADx) (x=0,1) | 206 |
| 12.5.3 CCPxA data register (CCPDxA) (x=0,1) | 206 |
| 12.5.4 CCPxB data register (CCPDxB) (x=0,1) | 207 |
| 12.5.5 CCP interrupt enable register (CCPIMSC) | 207 |

| | | |
|-------------------|--------------------------------------------------------------|------------|
| 12.5.6 | CCP interrupt source status register (CCPRIS)..... | 208 |
| 12.5.7 | CCP enabled interrupt status register (CCPMIS) | 209 |
| 12.5.8 | CCP interrupt clear register (CCPICLR) | 210 |
| 12.5.9 | CCP operation register (CCPRUN)..... | 210 |
| 12.5.10 | CCP write enable control register (LOCK) | 210 |
| 12.5.11 | CAP control register (CAPCON) | 211 |
| 12.5.12 | CAP channel select register (CAPCHS) | 213 |
| 12.5.13 | CAP data register (CAP0DATA) | 214 |
| 12.5.14 | CAP data register (CAPnDATA) (n=1-3)..... | 214 |
| Chapter 13 | Enhanced PWM (EPWM)..... | 215 |
| 13.1 | Overview | 215 |
| 13.2 | Features | 215 |
| 13.3 | Function description | 216 |
| 13.3.1 | Block diagram..... | 217 |
| 13.3.2 | Clock division..... | 217 |
| 13.3.3 | Independent output mode | 217 |
| 13.3.4 | Complementary output mode | 218 |
| 13.3.5 | Synchronous output mode | 218 |
| 13.3.6 | Group output mode..... | 218 |
| 13.3.7 | Load update mode..... | 219 |
| 13.3.8 | Edge-aligned counting mode..... | 222 |
| 13.3.9 | Center-aligned counting mode | 223 |
| 13.3.10 | Independent counter compare function..... | 226 |
| 13.3.11 | Programmable dead-time generator | 228 |
| 13.3.12 | Mask and mask preset function | 229 |
| 13.3.13 | Hall sensor interface function | 229 |
| 13.3.14 | Fault protection function (brake and recovery function)..... | 232 |
| 13.3.15 | Output status in debug mode | 234 |
| 13.3.16 | Output channel remapping | 234 |
| 13.3.17 | EPWM configuration process | 234 |
| 13.3.18 | Interrupts..... | 235 |
| 13.4 | Register mapping | 236 |
| 13.5 | Register description | 238 |
| 13.5.1 | EPWM prescaler register (CLKPSC) | 238 |
| 13.5.2 | EPWM clock selection register (CLKDIV) | 239 |
| 13.5.3 | EPWM control register (CON)..... | 241 |
| 13.5.4 | EPWM control register (CON2)..... | 243 |
| 13.5.5 | EPWM control register (CON3)..... | 244 |
| 13.5.6 | EPWM period register 0-5 (PERIOD0-5) | 247 |
| 13.5.7 | EPWM compare register 0-5(CMPDAT0-5)..... | 247 |
| 13.5.8 | EPWM output control register (POEN)..... | 247 |
| 13.5.9 | EPWM output channel remap register (POREMAP) | 248 |
| 13.5.10 | EPWM fault protection control register (BRKCTL)..... | 250 |
| 13.5.11 | EPWM dead time control register (DTCTL) | 253 |
| 13.5.12 | EPWM mask output control register (MASK)..... | 254 |
| 13.5.13 | EPWM mask output control preset register (MASKNXT) | 255 |
| 13.5.14 | EPWM trigger compare register (CMPTGD0-1) | 257 |

| | | |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------|------------|
| 13.5.15 | EPWM interrupt enable register (IMSC)..... | 258 |
| 13.5.16 | EPWM interrupt source status register (RIS)..... | 259 |
| 13.5.17 | EPWM enabled interrupt status register (MIS) | 260 |
| 13.5.18 | EPWM interrupt clear control register (ICLR) | 261 |
| 13.5.19 | EPWM interrupt accumulation control register (IFA) | 262 |
| 13.5.20 | EPWM write enable control register (LOCK) | 262 |
| 13.5.21 | EPWM fault protection recovery delay register (BRKRDT) | 263 |
| Chapter 14 Universal Asynchronous Receiver Transmitter (UART) | | 264 |
| 14.1 | Overview | 264 |
| 14.2 | Features | 264 |
| 14.3 | Function description | 264 |
| 14.3.1 | UART function mode | 264 |
| 14.3.2 | UART interrupts and status | 264 |
| 14.4 | Register mapping | 265 |
| 14.5 | Register description | 266 |
| 14.5.1 | Receive buffer register (RBR) | 266 |
| 14.5.2 | Transmit buffer register (THR) | 266 |
| 14.5.3 | Baud rate divider register (DLR) | 266 |
| 14.5.4 | Interrupt enable register (IER)..... | 266 |
| 14.5.5 | Interrupt status register (IIR) | 267 |
| 14.5.6 | Line control register (LCR) | 267 |
| 14.5.7 | Modem control register (MCR)..... | 268 |
| 14.5.8 | Line status register (LSR)..... | 268 |
| 14.5.9 | UART access end register (END) | 269 |
| Chapter 15 I²C Serial Interface Controller (I²C) | | 270 |
| 15.1 | Overview | 270 |
| 15.2 | Features | 270 |
| 15.3 | Register mapping | 271 |
| 15.4 | Register description | 272 |
| 15.4.1 | I ² C control set register (CONSET) | 272 |
| 15.4.2 | I ² C control clear register (CONCLR) | 273 |
| 15.4.3 | I ² C status register (STAT) | 274 |
| 15.4.4 | I ² C data register (DAT) | 275 |
| 15.4.5 | I ² C clock control register (CLK)..... | 275 |
| 15.4.6 | I ² C slave address register (ADR0/ADR1/ADR2/ADR3) | 275 |
| 15.4.7 | I ² C slave address mask register (ADM0/ADM1/ADM2/ADM3)..... | 275 |
| 15.4.8 | I ² C extended slave address register (XADR0)..... | 276 |
| 15.4.9 | I ² C extended slave address mask register (XADM0)..... | 276 |
| 15.4.10 | I ² C software reset register (RST) | 276 |
| Chapter 16 SPI Controller (SSP/SPI)..... | | 277 |
| 16.1 | Overview | 277 |
| 16.2 | Features | 277 |
| 16.3 | Register mapping | 277 |
| 16.4 | Register description | 278 |
| 16.4.1 | SSP control register (CON) | 278 |
| 16.4.2 | SSP status register (STAT) | 279 |

| | | |
|----------------------------------------------------------------|---------------------------------------------------------------------------------|------------|
| 16.4.3 | SSP data register (DAT)..... | 279 |
| 16.4.4 | SSP clock controller (CLK)..... | 279 |
| 16.4.5 | SSP interrupt enable register (IMSC)..... | 280 |
| 16.4.6 | SSP interrupt source status register (RIS)..... | 280 |
| 16.4.7 | SSP enabled interrupt status register (MIS) | 280 |
| 16.4.8 | SSP interrupt clear register (ICLR) | 281 |
| 16.4.9 | SSP software chip select signal register (CSCR) | 281 |
| Chapter 17 Analog-to-Digital Conversion (ADC) | | 282 |
| 17.1 | Overview | 282 |
| 17.2 | Features | 282 |
| 17.3 | Functional description | 283 |
| 17.3.1 | ADC channels..... | 283 |
| 17.3.2 | Block diagram of ADC structure | 284 |
| 17.3.3 | ADC power consumption modes..... | 285 |
| 17.3.4 | ADC conversion modes..... | 285 |
| 17.3.5 | ADC clock..... | 285 |
| 17.3.6 | ADC channel selection and interrupt generation | 286 |
| 17.3.7 | ADC software start | 286 |
| 17.3.8 | ADC hardware trigger start..... | 287 |
| 17.4 | Register mapping | 290 |
| 17.5 | Register description | 292 |
| 17.5.1 | ADC control register (CON)..... | 292 |
| 17.5.2 | ADC control register 2 (CON2)..... | 294 |
| 17.5.3 | ADC hardware trigger control register (HWTG)..... | 295 |
| 17.5.4 | ADC EPWM trigger delay register (EPWMTGDLY)..... | 296 |
| 17.5.5 | ADC scan register (SCAN)..... | 296 |
| 17.5.6 | ADC EPWM output trigger conversion channel enable register (CHEPWM)..... | 296 |
| 17.5.7 | ADC EPWM comparator 0 trigger conversion channel enable register (CHPTG0) | 297 |
| 17.5.8 | ADC EPWM comparator 1 trigger conversion channel enable register (CHPTG1) | 297 |
| 17.5.9 | ADC conversion result rRegister (DATAx) x=23-0,x≠19..... | 297 |
| 17.5.10 | ADC compare control register 0 (CMPx) x=0~1 | 298 |
| 17.5.11 | ADC interrupt enable register (IMSC) | 299 |
| 17.5.12 | ADC interrupt source status register (RIS) | 299 |
| 17.5.13 | ADC enabled interrupt status register (MIS) | 299 |
| 17.5.14 | ADC interrupt clear register (ICLR) | 300 |
| 17.5.15 | ADC write enable control register (LOCK) | 300 |
| Chapter 18 Programmable Gain Amplifier (PGA0/1/2) | | 301 |
| 18.1 | Overview | 301 |
| 18.2 | Features | 301 |
| 18.3 | Block diagram of structure | 302 |
| 18.4 | Register mapping | 303 |
| 18.5 | Register description | 304 |
| 18.5.1 | PGA0 control register 0 | 304 |
| 18.5.2 | PGA0 control register 1 | 304 |
| 18.5.3 | PGA0 access enable register..... | 305 |
| 18.5.4 | PGA1 control register 0 | 305 |

| | | |
|-------------------|------------------------------------------------------------------------|------------|
| 18.5.5 | PGA2 control register 0 | 306 |
| 18.5.6 | PGA12 control register 0 | 306 |
| 18.5.7 | PGA1/PGA2 access register enable | 306 |
| Chapter 19 | Analog Comparator (ACMP0/1)..... | 307 |
| 19.1 | Overview | 307 |
| 19.2 | Block diagram of structure | 307 |
| 19.3 | Features | 309 |
| 19.4 | Function description | 309 |
| 19.5 | Register mapping | 310 |
| 19.6 | Register description | 311 |
| 19.6.1 | Analog comparator 0 control register 0 (C0CON0)..... | 311 |
| 19.6.2 | Analog comparator 0 control register 1 (C0CON1)..... | 312 |
| 19.6.3 | Analog comparator 1 control register 0 (C1CON0)..... | 313 |
| 19.6.4 | Analog comparator 1 control register 1 (C1CON1)..... | 314 |
| 19.6.5 | Analog comparator event control register (CEVCON) | 315 |
| 19.6.6 | Analog comparator interrupt enable register (IMSC) | 315 |
| 19.6.7 | Analog comparator interrupt source status register (RIS) | 315 |
| 19.6.8 | Analog comparator enabled interrupt source status register (MIS) | 316 |
| 19.6.9 | Analog comparator interrupt clear control register (ICLR) | 316 |
| 19.6.10 | Analog comparator write enable control register (LOCK) | 316 |
| Chapter 20 | DAC | 317 |
| 20.1 | Overview | 317 |
| 20.2 | Block diagram of structure | 317 |
| 20.3 | Features | 317 |
| 20.4 | Register mapping | 317 |
| 20.5 | Register description | 318 |
| 20.5.1 | DAC control register 0(CON0)..... | 318 |
| 20.5.2 | DAC write enable control register (LOCK) | 318 |
| Chapter 21 | Overview of ADCLDO | 319 |
| 21.1 | Features | 319 |
| 21.2 | Block diagram of structure | 319 |
| 21.3 | Register mapping | 319 |
| 21.4 | Register description | 320 |
| 21.4.1 | ADCLDO control register 0(CON0) | 320 |
| 21.4.2 | ADCLDO write enable control register (LOCK)..... | 320 |
| Chapter 22 | Nested Vectored Interrupt Controller (NVIC) | 321 |
| 22.1 | Features | 321 |
| 22.2 | Exception mode and system interrupt mapping | 322 |
| 22.3 | Vector table | 323 |
| 22.4 | Register mapping | 324 |
| 22.5 | Register description | 325 |
| 22.5.1 | Interrupt set enable control register (ISER) | 325 |
| 22.5.2 | Interrupt clear enable control register (ICER) | 325 |
| 22.5.3 | Interrupt set pending control register (ISPR) | 325 |
| 22.5.4 | Interrupt clear pending control register (ICPR) | 326 |

| | | |
|-------------------|--------------------------------------------------------------|------------|
| 22.5.5 | IRQ0~IRQ3 Interrupt priority register (IPR0)..... | 326 |
| 22.5.6 | IRQ4~IRQ7 Interrupt priority register (IPR1)..... | 326 |
| 22.5.7 | IRQ8~IRQ11 Interrupt priority register (IPR2) | 327 |
| 22.5.8 | IRQ12~IRQ15 Interrupt priority register (IPR3) | 328 |
| 22.5.9 | IRQ16~IRQ19 Interrupt priority register (IPR4) | 328 |
| 22.5.10 | IRQ20~IRQ23 Interrupt priority register (IPR5) | 329 |
| 22.5.11 | IRQ24~IRQ27 Interrupt priority register (IPR6) | 329 |
| 22.5.12 | IRQ28~IRQ31 Interrupt priority register (IPR7) | 330 |
| 22.5.13 | External interrupt rising edge enable register (EGP0) | 331 |
| 22.5.14 | External interrupt falling edge enable register (EGN0) | 332 |
| Chapter 23 | Standby Function | 333 |
| 23.1 | Standby function..... | 333 |
| 23.2 | Sleep mode | 333 |
| 23.2.1 | Setting of sleep mode..... | 333 |
| 23.2.2 | Sleep mode release..... | 335 |
| 23.3 | Deep sleep mode | 337 |
| 23.3.1 | Setting of deep sleep mode..... | 337 |
| 23.3.2 | Deep sleep mode release | 339 |
| 23.4 | Deep sleep mode with partial power down | 340 |
| 23.4.1 | Setting of deep sleep mode with partial power down | 340 |
| 23.4.2 | Release deep sleep mode with partial power down | 343 |
| Chapter 24 | Reset Function | 344 |
| 24.1 | Registers for confirming the reset source | 350 |
| 24.1.1 | Register mapping | 350 |
| 24.1.2 | Reset control flag register (RESF) | 350 |
| Chapter 25 | Power-On-Reset Circuit..... | 353 |
| 25.1 | Function of power-on-reset circuit..... | 353 |
| 25.2 | Structure of power-on reset circuit..... | 354 |
| 25.3 | Operation of power-on reset circuit..... | 355 |
| Chapter 26 | Voltage Detection Circuit | 359 |
| 26.1 | Function of voltage detection circuit..... | 359 |
| 26.2 | Structure of voltage detection circuit..... | 361 |
| 26.3 | Register mapping | 361 |
| 26.4 | Registers for controlling voltage detection circuit | 362 |
| 26.4.1 | Voltage detection register (LVIM) | 362 |
| 26.4.2 | Voltage detection level register (LVIS) | 363 |
| 26.5 | Operation of voltage detection circuit | 364 |
| 26.5.1 | When used as reset mode | 364 |
| 26.6 | When used as interrupt mode | 365 |
| 26.7 | When used as interrupt & reset mode | 367 |
| 26.8 | Cautions for voltage detection circuit..... | 373 |
| Chapter 27 | Safety Functions | 375 |
| 27.1 | Overview of safety functions | 375 |
| 27.2 | Register mapping | 375 |
| 27.3 | Operation of safety functions | 377 |

| | | |
|-------------------|----------------------------------------------------------------------------|------------|
| 27.3.1 | Flash CRC operation function (high-speed CRC)..... | 377 |
| 27.3.1.1 | Flash memory CRC operation result register L (PGCRCL) | 378 |
| 27.3.1.2 | CRC operation function (general-purpose CRC) | 380 |
| 27.3.1.3 | CRC input register (CRCIN) | 381 |
| 27.3.1.4 | CRC data register (CRCD) | 382 |
| 27.3.2 | SFR guard function..... | 383 |
| 27.3.2.1 | SFR guard control register (SFRGD) | 383 |
| 27.3.3 | Frequency detection function | 384 |
| 27.3.3.1 | Timer input/output select register 0 (TIOS0) | 384 |
| 27.3.4 | A/D test function | 385 |
| 27.3.4.1 | ADC register | 385 |
| 27.3.5 | Digital output signal level detection function for input/output pins | 386 |
| 27.3.5.1 | Port mode select register (PMS) | 386 |
| 27.3.6 | Product unique ID register..... | 387 |
| Chapter 28 | Temperature Sensor | 388 |
| 28.1 | Function of temperature sensor | 388 |
| 28.2 | Register mapping | 388 |
| 28.3 | Temperature sensor register..... | 388 |
| 28.3.1 | Temperature sensor calibration data register TSN25 | 388 |
| 28.4 | Instructions for using temperature sensor | 389 |
| Chapter 29 | Option Byte..... | 390 |
| 29.1 | Function of option byte | 390 |
| 29.1.1 | User option bytes (000C0H~000C2H) | 390 |
| 29.2 | Flash memory data protection option bytes (000C3H, 500004H) | 391 |
| 29.3 | Register mapping | 391 |
| 29.4 | User option bytes | 392 |
| 29.4.1 | User option byte (000C0H)..... | 392 |
| 29.4.2 | User option byte (000C1H)..... | 393 |
| 29.4.3 | User option byte (000C2H)..... | 396 |
| 29.4.4 | Flash memory data protection option byte (000C3H) | 396 |
| 29.4.5 | Flash memory data protection option byte (500004H)..... | 396 |
| Chapter 30 | FLASH Control | 397 |
| 30.1 | Overview of FLASH control..... | 397 |
| 30.2 | Structure of FLASH memory | 397 |
| 30.3 | Register mapping | 398 |
| 30.4 | Register description | 399 |
| 30.4.1 | Flash write protection register (FLPROT) | 399 |
| 30.4.2 | FLASH operation control register (FLOPMD1) | 399 |
| 30.4.3 | FLASH operation control register (FLOPMD2) | 399 |
| 30.4.4 | Flash erase control register (FLERMD) | 400 |
| 30.4.5 | Flash status register (FLSTS) | 400 |
| 30.4.6 | Flash chip erase time control register (FLCERCNT) | 401 |
| 30.4.7 | Flash sector erase time control register (FLSERCNT) | 401 |
| 30.4.8 | Flash write time control register (FLPROCNT) | 402 |
| 30.5 | How to operate FLASH | 403 |
| 30.5.1 | Sector erase | 403 |

| | | |
|----------------------------------------|------------------------------------|------------|
| 30.5.2 | Chip erase | 404 |
| 30.5.3 | Word program..... | 404 |
| 30.6 | Flash memory read | 405 |
| 30.7 | Cautions for FLASH operation | 405 |
| Appendix Revision History | | 406 |

Chapter 1 CPU

1.1 Overview

This chapter provides a brief introduction to the features and debugging features of the ARM Cortex-M0+ core. For details, please refer to the ARM related documentation.

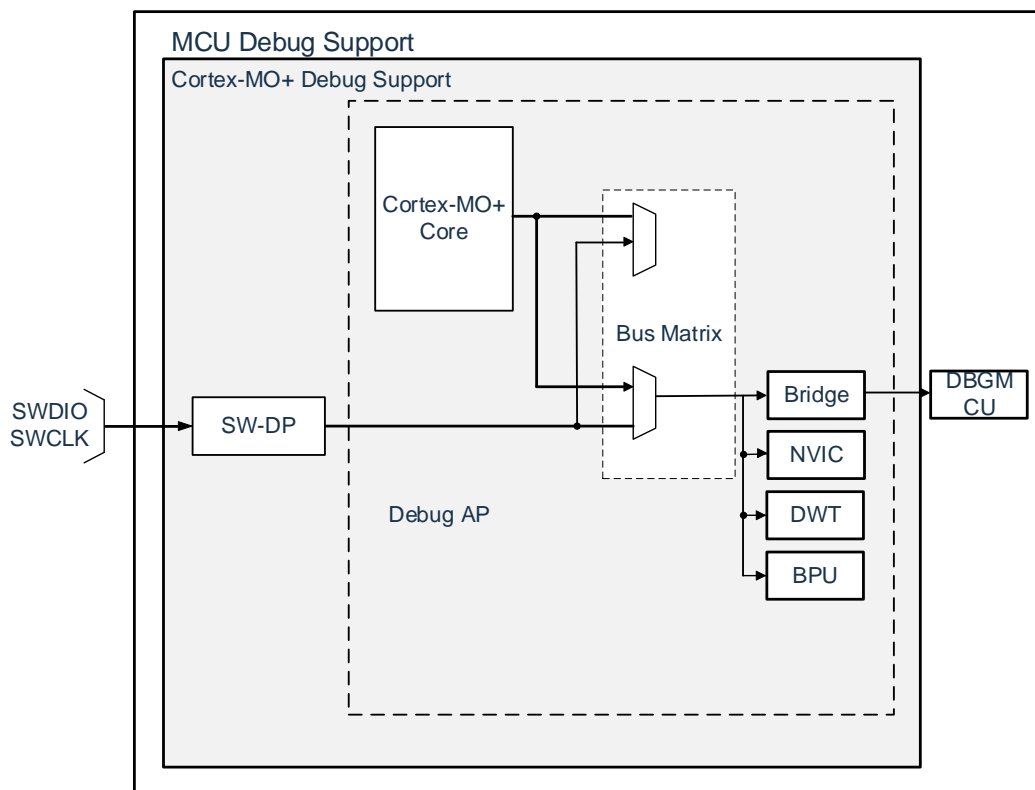
1.2 Cortex-M0+ core features

- ARM Cortex-M0+ processor is a 32-bit RISC core with a 2-stage pipeline that supports privileged mode only.
- 1-cycle hardware multiplier
- Nested vector interrupt controller (NVIC)
 - 1 non-maskable interrupt (NMI)
 - Support 21 maskable interrupt requests (IRQ)
 - 4 interrupt priority levels
- System Timer (SysTick) is a 24-bit countdown timer with a choice of F_{CLK} or F_{IL} count clock
- Vector table offset register (VTOR)
 - The software can write VTOR to relocate the vector table start address to a different location.
 - The default value of this register is 0x0000_0000, the lower 8 bits are ignored for writing and zero for reading, which means the offset is 256 bytes aligned.

Chapter 2 Debugging Features

- 2-wire SWD debug interface
- Support for suspending, resuming and single-step execution of programs
- Access to the processor's core registers and special function registers
- 4 hardware breakpoints (BPU)
- Unlimited software breakpoints (BKPT instruction)
- 2 data observation points (DWT)
- Accessing memory while the core is executing

Figure 2-1: Debug block diagram of Cortex-M0+



Note: SWD does not work in deep sleep mode, please do debug operation in active and sleep mode.

2.1 SWD interface pins

The 2 GPIOs of this product can be used as SWD interface pins, which exist in all packages.

Table 2-1: SWD debug port pins

| SWD port name | Debugging function | Pin assignment |
|---------------|--------------------------|----------------|
| SWDCLK | Serial clock | P06 |
| SWDIO | Serial data input/output | P07 |

When the SWD function is not used, SWD can be disabled by setting the debug stop control register (DBGSTOPCR).

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:25 | - | Reserved | - |
| 24 | SWDIS | SWD debug interface status 0: Enable the SWD debug interface. P06/P07 cannot be used as GPIO (because ENO and DOUT of this IOBUF are controlled by the debugger at this time). 1: Disable the SWD debug interface. P06/P07 can be used as GPIO. | 0 |
| 23:2 | - | Reserved | 0x0 |
| 1 | FRZEN1 | When the debugger is connected and the CPU is in debug state (HALTED=1), the peripheral module of the communication system acts/stops ^{Note 1} 0: Peripheral acts 1: Peripheral stops | 0 0 0 |
| 0 | FRZEN0 | When the debugger is connected and the CPU is in debug state (HALTED=1), the timer system peripheral module acts/stops ^{Note 2} 0: Peripheral acts 1: Peripheral stops | 0 0 0 |

Note 1: The communication system peripheral module of this product includes: serial communication unit, serial IICA.

Note 2: The timer system peripheral module of this product includes: general-purpose timer unit Timer4.

2.2 ARM reference documents

The built-in debugging feature in the Cortex®-M0+ core is part of the ARM® CoreSight design suite. For documentation, refer to:

- Cortex®-M0+ Technical Reference Manual (TRM)
- ARM® Debug Interface V5
- ARM® CoreSight Design Suite Version r1p1 Technical Reference Manual

Chapter 3 Pin Function

3.1 Port function

Refer to the datasheet of the corresponding product series for specific port functions.

3.2 Port multiplexing function

The specific port multiplexing functions are described in the datasheets for each product family. See the table below for details of the port multiplexing functions.

Table 3-1: Port multiplexing function digital mapping table

| Function name | Input | Multiplexing function PmnCFG | | | | | |
|---------------|-------------------------|------------------------------|---------|---------|---------|---------|-----|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| P00 | BKIN | Default multiplexing Output | - | - | - | - | - |
| P01 | - | | - | - | - | - | - |
| P02 | CCPnA_I / CCPnB_I | | C1_O | SSIO | CCP0A_O | CCP1A_O | - |
| P03 | RXD | | -- | SCK | | | SDA |
| P04 | CCPnA_I / CCPnB_I | | TXD | MISO | CCP0B_O | CCP1B_O | SCL |
| P05 | | | C0_O | MOSI | - | - | - |
| P06 | CCPnA_I / CCPnB_I / RXD | | - | - | CCP0A_O | CCP1A_O | - |
| P07 | CCPnA_I / CCPnB_I | | TXD | | CCP0B_O | CCP1B_O | |
| P10 | - | | EPWM0 | - | - | - | - |
| P11 | - | | EPWM1 | - | - | - | - |
| P12 | - | | EPWM2 | - | - | - | - |
| P13 | - | | EPWM3 | - | - | - | - |
| P14 | - | | EPWM4 | - | - | - | - |
| P15 | - | | EPWM5 | - | - | - | - |
| P16 | - | | EPWM0 | - | - | - | - |
| P20 | - | | - | - | - | - | - |
| P21 | - | | - | - | - | - | - |
| P22 | - | | - | - | - | - | - |
| P23 | - | | CCP0A_O | CCP1A_O | - | - | - |
| P24 | BKIN | | CCP0B_O | CCP1B_O | - | - | - |
| P25 | - | | - | - | - | - | - |
| P26 | CCPnA_I / CCPnB_I /BKIN | | C1_O | - | CCP0A_O | CCP1A_O | |
| P27 | - | | - | - | - | - | - |

Note 1: This product requires users to configure PMC, PM, and other registers separately for the IO multiplexing function.

Note 2: When selecting the IIC function, the open-drain function is automatically enabled.

Note 3: Regarding the multiplexing function, it can be used for both input and output. Once the PmnCFG is selected, the input channel is automatically enabled.

Table 3-2: Analog function and special function pins

| Pin | Analog | | | Special function pin |
|-----|--------|-------|------|----------------------|
| | ADC | PGA | ACMP | |
| P00 | AN8 | A0P | C1P3 | |
| P01 | AN9 | A0GND | | |
| P02 | AN10 | A0O | | RESINB |
| P03 | AN11 | | | |
| P04 | AN12 | A12O | | |
| P05 | AN13 | | | |
| P06 | | | | SWDCLK |
| P07 | | | | SWDIO |
| P10 | | | | |
| P11 | | | | |
| P12 | | | | |
| P13 | | | | |
| P14 | | | | |
| P15 | | | | |
| P16 | AN18 | | | |
| P20 | AN4 | | C0P0 | |
| P21 | AN5 | | C0P1 | |
| P22 | AN6 | | C0P2 | |
| P23 | AN7 | | C0N | |
| P24 | AN14 | A1P | C1P0 | |
| P25 | AN15 | A1GND | C1P1 | |
| P26 | AN16 | A2P | C1P2 | |
| P27 | AN17 | A2GND | C1N | |

3.3 Register mapping

3.3.1 Control function register mapping

(Base address of the port control register=0x40040000) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| P0 | 0x000 | R/W | Set the register for configuring the output latch value in 1-bit units; read this register in input mode to get the pin level, and in output mode to get the value of the port's output latch. | 0x00 |
| P1 | 0x001 | R/W | | 0x00 |
| P2 | 0x002 | R/W | | 0x00 |
| PM0 | 0x020 | R/W | When the port is used as a digital channel, the registers for the input or output of the port are set in 1-bit units. | 0xFF |
| PM1 | 0x021 | R/W | | 0xFF |
| PM2 | 0x022 | R/W | | 0xFF |
| PU0 | 0x030 | R/W | The internal pull-up resistor selection register of the port can only be set when the corresponding PMCmn=0, the pull-up resistor is valid. The pull-up function of P02 is enabled by default after a reset signal is generated. | 0x04 |
| PU1 | 0x031 | R/W | | 0x00 |
| PU2 | 0x032 | R/W | | 0x00 |
| PD0 | 0x040 | R/W | The internal pull-down resistor selection register of the port can only be set when the corresponding PMCmn=0, the pull-down resistor is valid. The pull-down function of P06 and P07 is enabled by default after a reset signal is generated. RESINB(P02) port has no pull-down function, and bit2 of PD0 is invalid. | 0xC0 |
| PD1 | 0x041 | R/W | | 0x00 |
| PD2 | 0x042 | R/W | | 0x00 |
| POM0 | 0x050 | R/W | Open Drain Mode Register, N-Channel Open Drain will be turned on only when the port is configured for Output Mode. When P03CFG=0X05 or P04CFG=0X05, it will force P03 or P04 to turn on the open drain mode. | 0x00 |
| POM1 | 0x051 | R/W | | 0x00 |
| POM2 | 0x052 | R/W | | 0x00 |
| PMC0 | 0x060 | R/W | Port Mode Register, sets the port to be used as a digital or analog channel in 1-bit units; P00, P01, P02, P06, P07 are used as digital channels by default. | 0x38 |
| PMC1 | 0x061 | R/W | | 0xFF |
| PMC2 | 0x062 | R/W | | 0xFF |
| PSET0 | 0x070 | W | Sets the registers of the port output latch in 1-bit units. | 0x00 |
| PSET1 | 0x071 | W | | 0x00 |
| PSET2 | 0x072 | W | | 0x00 |
| PCLR0 | 0x080 | W | Clear the port output latch registers in 1- | 0x00 |

| | | | | |
|-------|-------|---|------------|------|
| PCLR1 | 0x081 | W | bit units. | 0x00 |
| PCLR2 | 0x082 | W | | 0x00 |

3.3.2 Output-input multiplexing function register mapping

(Base address of the output-input multiplexing function registers =0x40040800)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|-----------------------------|--------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| P00CFG | 0x00 | R/W | Port output multiplexing configuration register allows mapping the output functionality of peripheral modules to the corresponding ports. For specific pin function digit mapping, please refer to Table 3-1. The reset value of the port output multiplexing configuration register is 0x00, which corresponds to the default multiplexing function and GPIO function of the port. | 0x00 |
| P01CFG | 0x02 | R/W | | 0x00 |
| P02CFG | 0x04 | R/W | | 0x00 |
| P03CFG | 0x06 | R/W | | 0x00 |
| P04CFG | 0x08 | R/W | | 0x00 |
| P05CFG | 0x0A | R/W | | 0x00 |
| P06CFG | 0x0C | R/W | | 0x00 |
| P07CFG | 0x0E | R/W | | 0x00 |
| P10CFG | 0x10 | R/W | | 0x00 |
| P11CFG | 0x12 | R/W | | 0x00 |
| P12CFG | 0x14 | R/W | | 0x00 |
| P13CFG | 0x16 | R/W | | 0x00 |
| P14CFG | 0x18 | R/W | | 0x00 |
| P15CFG | 0x1A | R/W | | 0x00 |
| P16CFG | 0x1C | R/W | | 0x00 |
| P20CFG | 0x20 | R/W | | 0x00 |
| P21CFG | 0x22 | R/W | | 0x00 |
| P22CFG | 0x24 | R/W | | 0x00 |
| P23CFG | 0x26 | R/W | | 0x00 |
| P24CFG | 0x28 | R/W | | 0x00 |
| P25CFG | 0x2A | R/W | | 0x00 |
| P26CFG | 0x2C | R/W | | 0x00 |
| P27CFG | 0x2E | R/W | | 0x00 |
| PS _{int0} _CFG | 0x60 | R/W | External interrupt 0 input port multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{int1} _CFG | 0x61 | R/W | External interrupt 1 input port multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{int2} _CFG | 0x62 | R/W | External interrupt 2 input port multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{int3} _CFG | 0x63 | R/W | External interrupt 3 input port multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{tau0tin0} _CFG | 0x64 | R/W | TAU0 external input channel 0 port input multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{tau0tin1} _CFG | 0x65 | R/W | TAU0 external input channel 1 port input multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{tau0tin2} _CFG | 0x66 | R/W | TAU0 external input channel 2 port input multiplexing register, which can be mapped to any port. | 0x3f |
| PS _{tau0tin3} _CFG | 0x67 | R/W | TAU0 external input channel 3 port input multiplexing register, which can be mapped to any port. | 0x3f |

| | | | | |
|-----------------------------|------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PS _{uart0rx} _CFG | 0x68 | R/W | UART rxd signal input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| PS _{epwmnkin} _CFG | 0x69 | R/W | EPWM external brake input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| PS _{ccp0ain} _CFG | 0x6A | R/W | CCP0 channel a input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| PS _{ccp0bin} _CFG | 0x6B | R/W | CCP0 channel b input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| PS _{ccp1ain} _CFG | 0x6C | R/W | CCP1 channel a input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| PS _{ccp1bin} _CFG | 0x6D | R/W | CCP1 channel b input port multiplexing register, which can be mapped to a specific port. For specific mappings, please refer to Table 3-1. | 0x07 |
| P0TTLCFG | 0x70 | R/W | P0 port input level selection register. | 0x00 |
| P2TTLCFG | 0x72 | R/W | P2 port input level selection register. | 0x00 |
| PMS | 0x7B | R/W | Port read mode selection register; when the port is in output mode, this register selects the value of the port latch or the pin output level. See Section 27.3 for details. | 0x00 |

3.3.3 Special function port RESINB control register mapping

(Register base address =0x40020400) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-------------------------------------------------------------------------------|-------------|
| RSTM | 0x0B | R/W | Select the RESINB (P02) port as the external reset port or GPIO port register | 0x00 |

3.4 Register description

The port is controlled via the following registers.

- (1) Port register (Px)
- (2) Port mode register (PMx)
- (3) Pull-up resistor selection register (PUx)
- (4) Pull-down resistor selection register (PDx)
- (5) Port output mode register (POMx)
- (6) Port mode control register (PMCx)
- (7) Port set control register (PSETx)
- (8) Port clear control register (PCLR_x)
- (9) Port output multiplexing configuration register (PxxCFG)
- (10) Port input multiplexing configuration register (PSxx_CFG)
- (11) Port level selection register (PxTTLCFG)
- (12) Special function port RESINB control register (RSTM)

3.4.1 Port register (Px)

This is register Px (x=0 to 2) which sets the value of the port's output latch in 1-bit units. Reading this register in input mode gives the pin level, and reading it in output mode gives the value of the port's output latch. After a reset signal is generated, the value of the register changes to "00H".

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | Px7 | Bit 7 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 6 | Px6 | Bit 6 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 5 | Px5 | Bit 5 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 4 | Px4 | Bit 4 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 3 | Px3 | Bit 3 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 2 | Px2 | Bit 2 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 1 | Px1 | Bit 1 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |
| 0 | Px0 | Bit 0 of the port x mode register 0: Output "0" in output mode; input low in input mode 1: Output "1" in output mode; input high in input mode | 0 |

Note: The initial value must be set for unassigned bits.

3.4.2 Port mode register (PMx)

When the port is used as a digital channel, this is the register PMx (x=0~2) that sets its input/output in bits. After a reset signal is generated, all ports default to the input state.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | PMx7 | Bit 7 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 6 | PMx6 | Bit 6 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 5 | PMx5 | Bit 5 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 4 | PMx4 | Bit 4 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 3 | PMx3 | Bit 3 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 2 | PMx2 | Bit 2 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 1 | PMx1 | Bit 1 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |
| 0 | PMx0 | Bit 0 of the port x mode register 0: Output mode (used as output port (output buffer ON)) 1: Input mode (used as input port (output buffer OFF)) | 1 |

Note 1: The initial value must be set for unassigned bits.

Note 2: The P17 port is invalid, and bit 7 of the PM1 remains set to 1.

3.4.3 Pull-up resistor selection register (PUx)

On-chip pull-up resistor selection register PUx (x=0 to 2). The pull-up resistor can only be set when the corresponding PMCx bit is equal to 0.

After a reset signal is generated, the pull-up function of the P02 port is turned on automatically, and the pull-up function of the other ports will not be turned on by default.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| 7 | PUx7 | Bit 7 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 6 | PUx6 | Bit 6 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 5 | PUx5 | Bit 5 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 4 | PUx4 | Bit 4 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 3 | PUx3 | Bit 3 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 2 | PUx2 | Bit 2 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | x=0, reset value is 1 x=1 or 2, reset value is 0 |
| 1 | PUx1 | Bit 1 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |
| 0 | PUx0 | Bit 0 of the Px pin on-chip pull-up resistor selection 0: No on-chip pull-up resistor is connected 1: Connect the on-chip pull-up resistor | 0 |

Note 1: The initial value must be set for unassigned bits.

Note 2: Port P17 is invalid,

Note 3: Bit7 of the PU1 remains set to 0.

3.4.4 Pull-down resistor selection register (PDx)

On-chip pull-down resistor selection register PDx (x=0 to 2). The pull-down resistor can only be set when the corresponding PMCx bit is equal to 0; RESINB (P02) port has no pull-down function.

After a reset signal is generated, the pull-down function of P06 and P07 ports will be turned on automatically, and the pull-down function of other ports will not be turned on by default.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| 7 | PDx7 | Bit 7 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | x=0, reset value is 1 x=1 or 2, reset value is 0 |
| 6 | PDx6 | Bit 6 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | x=0, reset value is 1 x=1 or 2, reset value is 0 |
| 5 | PDx5 | Bit 5 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | 0 |
| 4 | PDx4 | Bit 4 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | 0 |
| 3 | PDx3 | Bit 3 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | 0 |
| 2 | PDx2 | Bit 2 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor Note: The RESINB (P02) port does not have a pull-down function. | 0 |
| 1 | PDx1 | Bit 1 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | 0 |
| 0 | PDx0 | Bit 0 of the Px pin on-chip pull-down resistor selection 0: No on-chip pull-down resistor is connected 1: Connect the on-chip pull-down resistor | 0 |

Note 1: The initial value must be set for unassigned bits.

Note 2: Port P17 is invalid, and bit7 of the PD1 remains set to 0.

3.4.5 Port output mode register (POMx)

Port Output Mode Register POMx (x=0~2), will only be enabled if configured to output mode N-channel open drain. When P03CFG=0x05 or P04CFG=0x05, it will force the N-channel open drain mode of P03 or P04 to be turned on.

After a reset signal is generated, the value of the registers= changes to “00H”.

Note: For the bit that sets the N-channel open-drain output mode (POMmn=1), no on-chip pull-up resistor is connected.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|------------------------------------------------------------------------------------------------------------|-------------|
| 7 | POMx7 | Bit 7 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 6 | POMx6 | Bit 6 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 5 | POMx5 | Bit 5 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 4 | POMx4 | Bit 4 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 3 | POMx3 | Bit 3 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 2 | POMx2 | Bit 2 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 1 | POMx1 | Bit 1 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |
| 0 | POMx0 | Bit 0 of the Px pin output mode selection 0: Typical output mode 1: N-channel open-drain output mode | 0 |

Note: Port P17 is invalid, and bit7 of the POM1 remains set to 0.

3.4.6 Port mode control register (PMCx)

Port Mode Register (PMCx (x=0~2)), sets the port as a digital (input/output) or analog (input) channel in 1-bit units.

After a reset signal is generated, P00, P01, P02, P06, P07 are used as digital channels by default (PMC00, PMC01, PMC02, PMC06, PMC07 are reset to "0"), and the other ports are used as analog channels by default, i.e., the corresponding bit of PMCx is equal to 1.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| 7 | PMCx7 | Bit 7 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | x=0, reset value is 0 x=1 or 2, reset value is 1 |
| 6 | PMCx6 | Bit 6 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | x=0, reset value is 0 x=1 or 2, reset value is 1 |
| 5 | PMCx5 | Bit 5 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | 1 |
| 4 | PMCx4 | Bit 4 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | 1 |
| 3 | PMCx3 | Bit 3 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | 1 |
| 2 | PMCx2 | Bit 2 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | x=0, reset value is 0 x=1 or 2, reset value is 1 |
| 1 | PMCx1 | Bit 1 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | x=0, reset value is 0 x=1 or 2, reset value is 1 |
| 0 | PMCx0 | Bit 0 of the Px pin digital (inputs/outputs) or analog (inputs) 0: Digital inputs/outputs (multiplexing functions other than analog inputs) 1: Analog inputs | x=0, reset value is 0 x=1 or 2, reset value is 1 |

Note 1: The initial value must be set for unassigned bits.

Note 2: Port P17 is invalid, bit7 of the PMC1 remains set to 1.

3.4.7 Port set control register (PSETx)

This is a register that sets the port output latch PSETx (x=0 to 2) in 1-bit units. After a reset signal is generated, the value of the register changes to “00H”.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|--------|-------------------------------------------------------------------------------------|-------------|
| 7 | PSETx7 | Bit 7 of the Px pin set control 0: No operation 1: Corresponding Px7 set to 1 | 0 |
| 6 | PSETx6 | Bit 6 of the Px pin set control 0: No operation 1: Corresponding Px6 set to 1 | 0 |
| 5 | PSETx5 | Bit 5 of the Px pin set control 0: No operation 1: Corresponding Px5 set to 1 | 0 |
| 4 | PSETx4 | Bit 4 of the Px pin set control 0: No operation 1: Corresponding Px4 set to 1 | 0 |
| 3 | PSETx3 | Bit 3 of the Px pin set control 0: No operation 1: Corresponding Px3 set to 1 | 0 |
| 2 | PSETx2 | Bit 2 of the Px pin set control 0: No operation 1: Corresponding Px2 set to 1 | 0 |
| 1 | PSETx1 | Bit 1 of the Px pin set control 0: No operation 1: Corresponding Px1 set to 1 | 0 |
| 0 | PSETx0 | Bit 0 of the Px pin set control 0: No operation 1: Corresponding Px0 set to 1 | 0 |

Note 1: The initial value must be set for unassigned bits.

Note 2: Port P17 is invalid, bit7 of the PSET1 remains set to 0.

3.4.8 Port clear control register (PCLR_x)

This is a register that sets the port output latch PCLR_x (x=0 to 2) in 1-bit units. After a reset signal is generated, the value of the register changes to “00H”.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|---------------------------------------------------------------------------------------|-------------|
| 7 | PCLR _x 7 | Bit 7 of the Px pin clear control 0: No operation 1: Corresponding Px7 set to 0 | 0 |
| 6 | PCLR _x 6 | Bit 6 of the Px pin clear control 0: No operation 1: Corresponding Px6 set to 0 | 0 |
| 5 | PCLR _x 5 | Bit 5 of the Px pin clear control 0: No operation 1: Corresponding Px5 set to 0 | 0 |
| 4 | PCLR _x 4 | Bit 4 of the Px pin clear control 0: No operation 1: Corresponding Px4 set to 0 | 0 |
| 3 | PCLR _x 3 | Bit 3 of the Px pin clear control 0: No operation 1: Corresponding Px3 set to 0 | 0 |
| 2 | PCLR _x 2 | Bit 2 of the Px pin clear control 0: No operation 1: Corresponding Px2 set to 0 | 0 |
| 1 | PCLR _x 1 | Bit 1 of the Px pin clear control 0: No operation 1: Corresponding Px1 set to 0 | 0 |
| 0 | PCLR _x 0 | Bit 0 of the Px pin clear control 0: No operation 1: Corresponding Px0 set to 0 | 0 |

Note 1: The initial value must be set for unassigned bits.

Note 2: Port P17 is invalid, bit7 of the PCLR1 remains set to 0.

3.4.9 Port output multiplexing configuration register (PmnCFG)

The port multiplexing configuration register can map the output function of some peripheral modules to any port, see Table 3-1. If the reset value of the Port Output Multiplexing Configuration Register is "00H", and the port is defaulted to multiplexing function and GPIO function. The setting steps are as follows.

- (1) Set PMCmn=0 to select digital input/output.
- (2) Set PMmn=0 to set the output mode.
- (3) Set PmnCFG to select the reset function output of the corresponding pin.

Note: SDA, SCL and SPI of IIC can be used as both output and input. After setting PmnCFG, the input function will be enabled automatically, and there is no need to do anything other than the above three steps.

The register is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PmnCFG[2:0] | Pmn pin output multiplexing function 0x00: Pmn corresponding to output mapping table =0x00 0x01: Pmn corresponding to output mapping table =0x01 0x02: Pmn corresponding to output mapping table =0x02 0x03: Pmn corresponding to output mapping table =0x03 0x04: Pmn corresponding to output mapping table =0x04 0x05: Pmn corresponding to output mapping table =0x05 Other: Disable selecting | 0x0 |

3.4.10 Port input multiplexing configuration register (PSxx_CFG)

INTP0, INTP1, INTP2, INTP3, TI00, TI01, TI02, TI03 can be mapped to any GPIO inputs; RXD, BKIN, CCP0AIN, CCP0BIN, CCP1AIN, CCP1BIN can be mapped to the specified GPIO inputs, the specific mapping of the input functions, see Table 3-1; Since each function has multiple pin inputs, so when using the input multiplexing function, you need to select specific GPIO inputs, the setting procedure is as follows:

- (1) Set PMCmn = 0
- (2) Set PMmn=1
- (3) Set PSxx_CFG to select the corresponding pin input.

PSintp0_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PSintp0_CFG[5:0] | INTP0 selects the GPIO input 0x00: Select P00 as INTP0 input 0x01: Select P01 as INTP0 input 0x26: Select P26 as INTP0 input 0x27: Select P27 as INTP0 input Other: Input low level | 0x3F |

PSintp1_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PSintp1_CFG[5:0] | INTP1 selects the GPIO input 0x00: Select P00 as INTP1 input 0x01: Select P01 as INTP1 input 0x26: Select P26 as INTP1 input 0x27: Select P27 as INTP1 input Other: Input low level | 0x3F |

PSintp2_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PSintp2_CFG[5:0] | INTP2 selects the GPIO input 0x00: Select P00 as INTP2 input 0x01: Select P01 as INTP2 input 0x26: Select P26 as INTP2 input 0x27: Select P27 as INTP2 input Other: Input low level | 0x3F |

PSintp3_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PSintp3_CFG[5:0] | INTP3 selects the GPIO input 0x00: Select P00 as INTP3 input 0x01: Select P01 as INTP3 input 0x26: Select P26 as INTP3 input 0x27: Select P27 as INTP3 input Other: Input low level | 0x3F |

PStau0tin0_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PStau0tin0_CFG[5:0] | TI00 selects the GPIO input 0x00: Select P00 as TI00 input 0x01: Select P01 as TI00 input 0x26: Select P26 as TI00 input 0x27: Select P27 as TI00 input Other: TI00 input low level | 0x3F |

PStau0tin1_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PStau0tin1_CFG[5:0] | TI01 selects the GPIO input 0x00: Select P00 as TI01 input 0x01: Select P01 as TI01 input 0x26: Select P26 as TI01 input 0x27: Select P27 as TI01 input Other: TI01 input low level | 0x3F |

PStau0tin2_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PStau0tin2_CFG[5:0] | TI02 selects the GPIO input 0x00: Select P00 as TI02 input 0x01: Select P01 as TI02 input 0x26: Select P26 as TI02 input 0x27: Select P27 as TI02 input Other: TI02 input low level | 0x3F |

PStau0tin3_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:6 | -- | Reserved | -- |
| 5:0 | PStau0tin3_CFG [5:0] | TI03 selects the GPIO input 0x00: Select P00 as TI03 input 0x01: Select P01 as TI03 input 0x26: Select P26 as TI03 input 0x27: Select P27 as TI03 input Other: TI03 input low level | 0x3F |

PSuart0rxn_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSuart0rxn_CFG [2:0] | UART0_RXD selects the GPIO input 0x00: Select P03 as UART_RXD input 0x01: Select P06 as UART_RXD input Other: UART0_RXD input high level | 0x07 |

PSepwmbkin_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSepwmbkin_CFG [2:0] | EPWM_BKIN selects the GPIO input 0x00: Select P00 as EPWM_BKIN input 0x01: Select P24 as EPWM_BKIN input 0x02: Select P26 as EPWM_BKIN input Other: EPWM_BKIN input low level | 0x07 |

PSccp0ain_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSccp0a_i_CFG [2:0] | CCP0AIN selects the GPIO input 0x00: Select P02 as CCP0A_I input 0x01: Select P04 as CCP0A_I input 0x02: Select P06 as CCP0A_I input 0x03: Select P07 as CCP0A_I input 0x04: Select P26 as CCP0A_I input Other: CCP0AIN input low level | 0x7 |

PSccp0bin_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSccp0b_i_CFG [2:0] | CCP0BIN selects the GPIO input 0x00: Select P02 as CCP0BIN input 0x01: Select P04 as CCP0BIN input 0x02: Select P06 as CCP0BIN input 0x03: Select P07 as CCP0BIN input 0x04: Select P26 as CCP0BIN input Other: CCP0BIN input low level | 0x7 |

PSccp1ain_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSccp1ain_CFG [2:0] | CCP1AIN selects the GPIO input 0x00: Select P02 as CCP1AIN input 0x01: Select P04 as CCP1AIN input 0x02: Select P06 as CCP1AIN input 0x03: Select P07 as CCP1AIN input 0x04: Select P26 as CCP1AIN input Other: CCP1AIN input low level | 0x7 |

PSccp1bin_CFG is described as follows:

| Bit | Symbol | Description | Reset value |
|-----|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | -- | Reserved | -- |
| 2:0 | PSccp1bin_CFG [2:0] | CCP1BIN selects the GPIO input 0x00: Select P02 as CCP1BIN input 0x01: Select P04 as CCP1BIN input 0x02: Select P06 as CCP1BIN input 0x03: Select P07 as CCP1BIN input 0x04: Select P26 as CCP1BIN input Other: CCP1BIN input low level | 0x7 |

3.4.11 TTL and Schmitt input selection (PxTTLCFG,x=0 and 2)

The PxTTLCFG selection register is described as follows.

| Bit | Symbol | Description | Reset value |
|-----|--------|---------------------------------------------------------------|-------------|
| 7 | PxTTL7 | Px7 input level selection 0: Schmitt input 1: TTL input | 0 |
| 6 | PxTTL6 | Px6 input level selection 0: Schmitt input 1: TTL input | 0 |
| 5 | PxTTL5 | Px5 input level selection 0: Schmitt input 1: TTL input | 0 |
| 4 | PxTTL4 | Px4 input level selection 0: Schmitt input 1: TTL input | 0 |
| 3 | PxTTL3 | Px3 input level selection 0: Schmitt input 1: TTL input | 0 |
| 2 | PxTTL2 | Px2 input level selection 0: Schmitt input 1: TTL input | 0 |
| 1 | PxTTL1 | Px1 input level selection 0: Schmitt input 1: TTL input | 0 |
| 0 | PxTTL0 | Px0 input level selection 0: Schmitt input 1: TTL input | 0 |

3.4.12 Special function port RESINB description (RSTM)

This product power-on default RESINB (P02) is valid, if you need to use this port as GPIO, then you need to turn off the reset function through the register, the register description is as follows:

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------|-------------|
| 7: 1 | -- | Reserved to 0 | 0x0 |
| 0 | RSTM | Masking the RESINB pin for external reset functions 0: RESINB external reset pin 1: RESINB as GPIO pin | 0 |

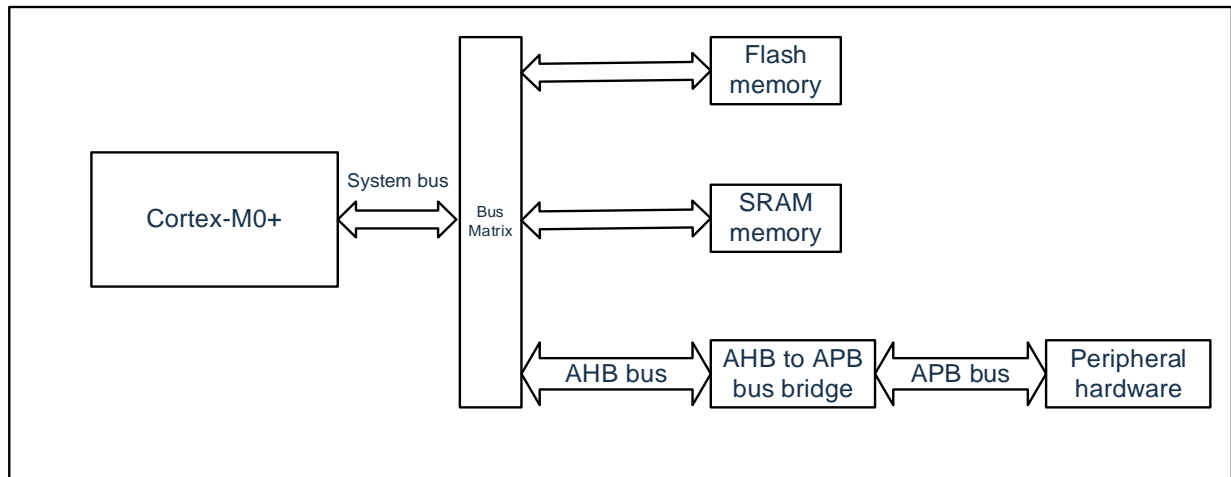
Chapter 4 System Architecture

4.1 Overview

This product system consists of the following components:

- 1 AHB bus Master:
 - Cortex-M0+
- 3 AHB buses Slaves:
 - FLASH memory
 - SRAM memory
 - AHB to APB Bridge, contains all APB interface peripherals

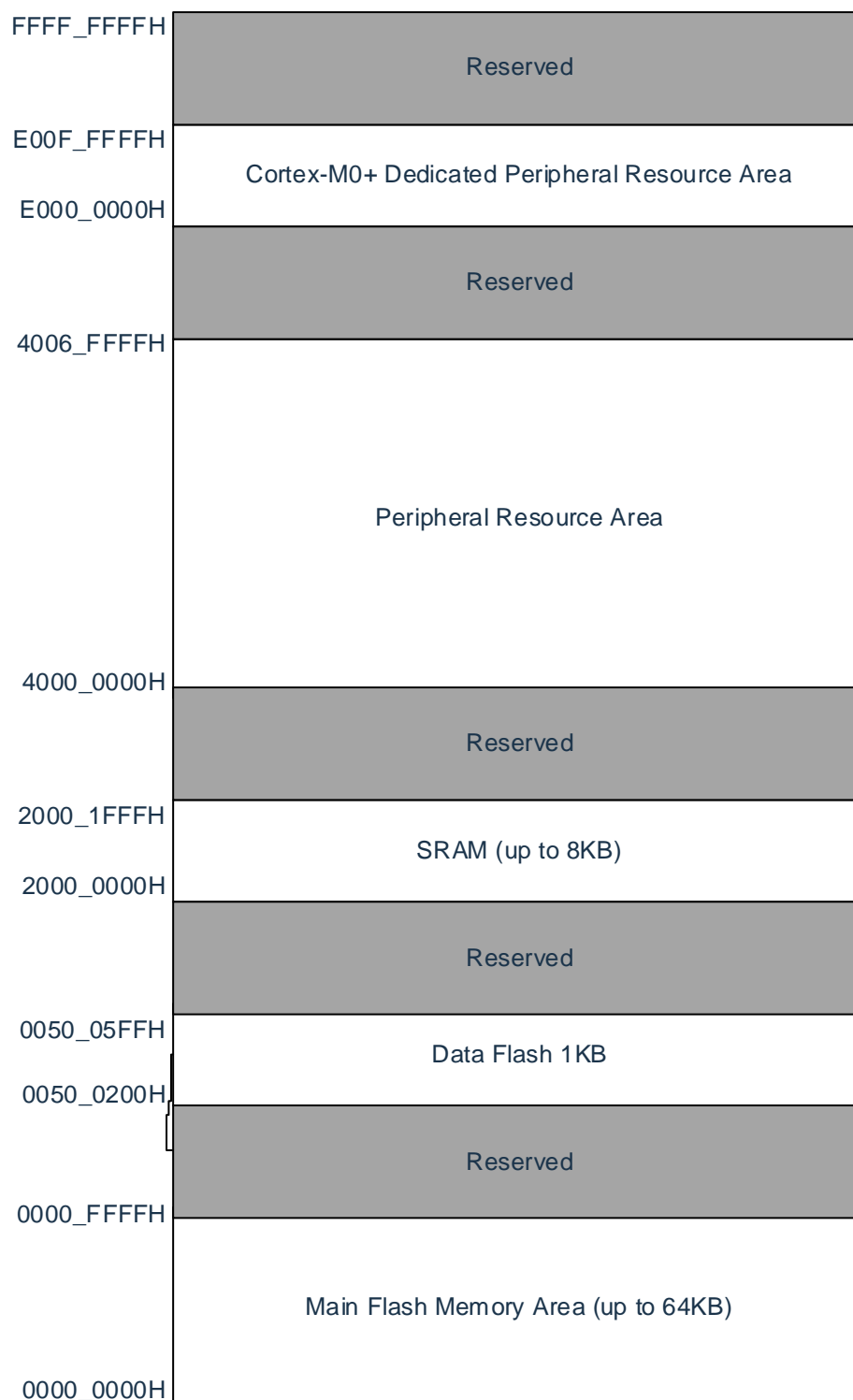
Figure 4-1: Block diagram of system architecture



- 1) System bus: This bus connects the system bus (peripheral bus) of the Cortex-M0+ core to the bus matrix.
- 2) Bus matrix: The bus matrix coordinates access to other buses on the core system bus.
- 3) AHB to APB Bridge: The AHB to APB Bridge provides a synchronous connection between the AHB and APB buses. Refer to Table 4-1 for address mapping of the different peripherals connected to each bridge.

4.2 System address partitioning

Figure 4-2: Map of address area



Peripheral Address Assignment

Table 4-1: Start address of peripheral register group

| Start address | Peripheral | Remark |
|---------------------------|----------------------------|--------------------------------|
| 0x4000_0000 - 0x4000_4FFF | Reserved | |
| 0x4000_5000 - 0x4000_5FFF | Reserved | |
| 0x4000_6000 - 0x4000_6FFF | Interrupt control | |
| 0x4000_7000 - 0x4001_8FFF | Reserved | |
| 0x4001_9000 - 0x4001_9FFF | Reserved | |
| 0x4001_A000 - 0x4001_FFFF | Reserved | |
| 0x4002_0000 - 0x4002_03FF | FLASH control | |
| 0x4002_0400 - 0x4002_0FFF | Clock control | |
| 0x4002_1000 - 0x4002_1001 | Watchdog timer | |
| 0x4002_1002 - 0x4002_1800 | Reserved | |
| 0x4002_1800 - 0x4002_1BFF | High-speed CRC | See Chapter 27 Safety Function |
| 0x4002_1C00 - 0x4002_1FFF | Clock control | |
| 0x4002_2000 - 0x4003_FFFF | Reserved | |
| 0x4004_0000 - 0x4004_0FFF | GPIO | |
| 0x4004_1D00 - 0x4004_1FFF | Timer array 0 | |
| 0x4004_2200 - 0x4004_23FF | Reserved | |
| 0x4004_2800 - 0x4004_31FF | Reserved | |
| 0x4004_3200 - 0x4004_32FF | General-purpose CRC | See Chapter 27 Safety Function |
| 0x4004_3300 - 0x4004_33FF | Reserved | |
| 0x4004_3C00 - 0x4004_3FFF | Reserved | |
| 0x4004_4000 - 0x4004_43FF | Reserved | |
| 0x4004_4B50 - 0x4004_4B50 | LSITIMER | |
| 0x4004_4800 - 0x4004_4EFF | Reserved | |
| 0x4004_5400 - 0x4004_5AFF | Reserved | |
| 0x4004_5B00 - 0x4004_5BFF | External interrupt control | |
| 0x4008_0000 - 0x4008_01FF | Reserved | |
| 0x4008_0200 - 0xDFFF_FFFF | Reserved | |
| - | | |
| 0x4006_1000 - 0x4006_1FFF | TIMER0 | |
| 0x4006_2000 - 0x4006_2FFF | Reserved | |
| 0x4006_3000 - 0x4006_3FFF | SPI | |
| 0x4006_4000 - 0x4006_40FF | UART | |
| 0x4006_4100 - 0x4006_41FF | Reserved | |
| 0x4006_4200 - 0x4006_427F | EPWM | |
| 0x4006_4280 - 0x4006_42FF | CCP | |
| 0x4006_4300 - 0x4006_433F | IIC | |
| 0x4006_4340 - 0x4006_437F | Reserved | |

| | | |
|---------------------------|----------|--|
| 0x4006_4380 – 0x4006_43BF | DIV | |
| | Reserved | |
| 0x4006_8000 – 0x4006_80FF | ADC | |
| 0x4006_8100 – 0x4006_81FF | Reserved | |
| 0x4006_8200 – 0x4006_823F | ACMP0 | |
| 0x4006_8240 – 0x4006_827F | Reserved | |
| - | Reserved | |
| 0x4006_8300 – 0x4006_831F | PGA0 | |
| 0x4006_8320 – 0x4006_833F | PGA12 | |
| 0x4006_8340 – 0x4006_835F | ADCLDO | |
| 0x4006_8360 – 0x4006_837F | DAC | |
| - | | |
| - | | |

Chapter 5 Clock Generation Circuit

5.1 Function of clock generation circuit

The clock generation circuit is a circuit that generates a clock supplied to the CPU and peripheral hardware. There are the following 2 types of system clock and clock oscillation circuits

(1) Main system clock high-speed on-chip oscillator (high-speed OCO)

The frequency at which to oscillate can be selected from among $F_{HOCO}=64\text{MHz}$, 48MHz , 32MHz , 24MHz , 16MHz , 12MHz , 8MHz , 6MHz , 4MHz , 3MHz and 2MHz (Typ.) by using the option byte (000C2H). After the reset is released, the CPU must start operation with this high-speed on-chip oscillator clock. Oscillation can be stopped by entering deep sleep mode or by setting the HIOSTOP bit (bit0 of the CSC register). The frequency set by the option byte can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV). Refer to “Setting of high-speed on-chip oscillator frequency select register (HOCODIV)” for frequency settings.

(2) Low-speed on-chip oscillator clock (low-speed OCO)

This circuit oscillates a clock of $F_{IL}=15\text{kHz}$.

The low-speed on-chip oscillator clock can be used as the system clock.

The low-speed on-chip oscillator oscillates when bit4 (WDTON) of the option byte (000C0H) or bit4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC) is “1” or when bit0 (SELLOSC) of the subsystem clock select register (SUBCKSEL) is “1”.

However, the low-speed on-chip oscillator stops oscillating if the deep sleep mode or sleep mode is entered when the WDTON bit is “1”, the WUTMMCK0 bit is “0”, and bit0 (WDSTBYON) of the option byte (000C0H) is “0”.

Note: F_{HOCO} : High-speed on-chip oscillator clock frequency

F_{IH} : High-speed on-chip oscillator clock frequency

F_{IL} : Low-speed on-chip oscillator clock frequency

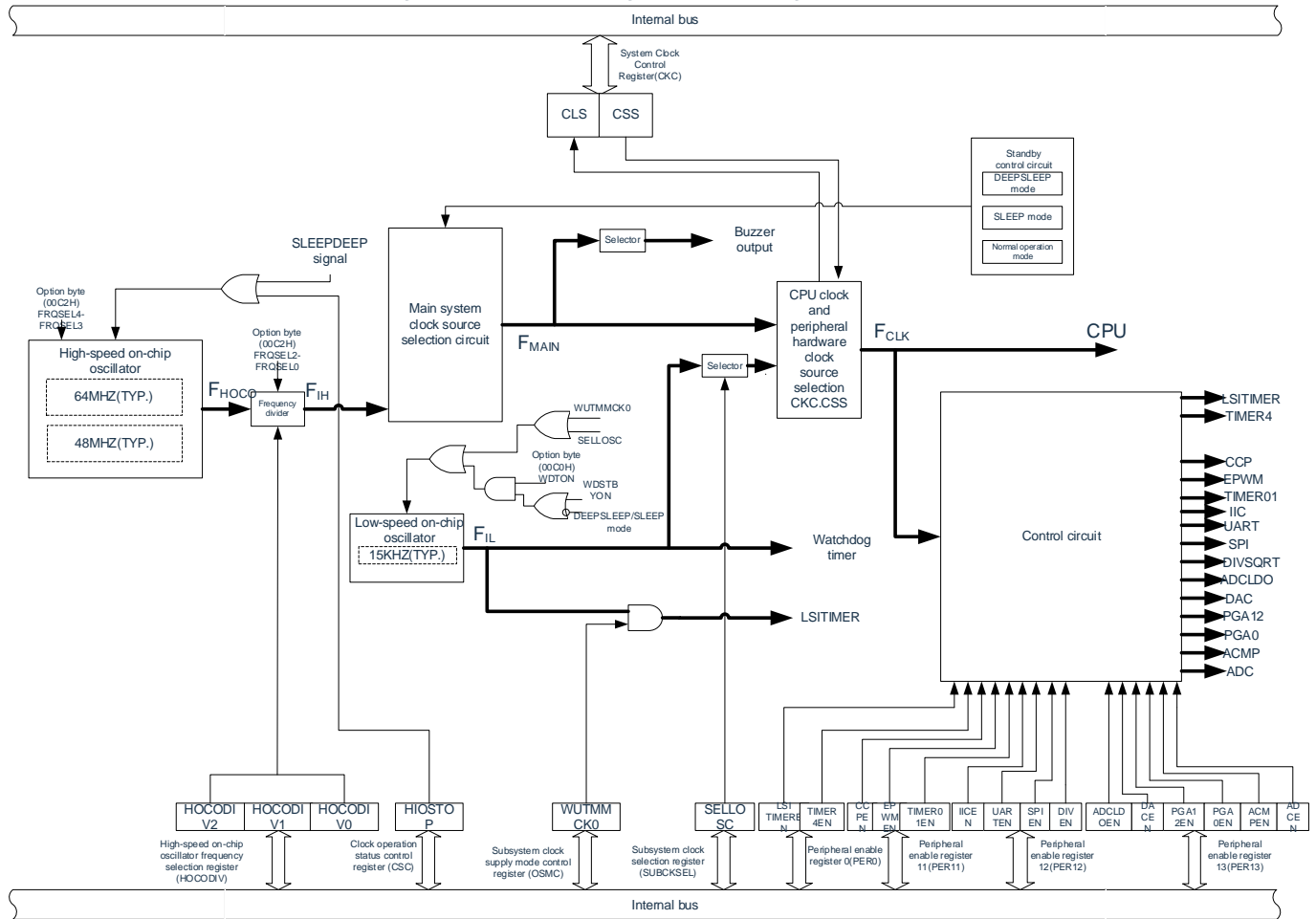
5.2 Configuration of clock generation circuit

The clock generation circuit includes the following hardware.

Table 5-1: Configuration of clock generation circuit

| Item | Configuration |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Control registers | System clock control register (CKC) Clock operation status control register (CSC) Peripheral enable registers 0, 1 (PER0, PER11, PER12, PER13) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) Subsystem clock select register (SUBCKSEL) |
| Oscillator circuits | High-speed on-chip oscillator Low-speed on-chip oscillator |

Figure 5-1: Block diagram of clock generation circuit



Note: F_{HOCO}: High-speed on-chip oscillator clock frequency

F_{IH}: High-speed on-chip oscillator clock frequency

F_{MAIN}: Main system clock frequency

F_{CLK}: CPU/peripheral hardware clock frequency

F_{IL}: Low-speed on-chip oscillator clock frequency

5.3 Register mapping

(Base address of the following registers = 0x4002_0400)

RO: Read Only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-----------------------------------------------|-------------|
| CSC | 0x001 | R/W | Clock operation state control register | 0xC0 |
| CKC | 0x004 | R/W | System clock control register | 0x0 |
| SUBCKSEL | 0x007 | R/W | Sub-system clock selection register | 0x0 |
| PMUKEY | 0x008 | R/W | Power mode control protection register | 0x0 |
| PMCCTL | 0x00A | R/W | Power mode control register | 0x0 |
| PER0 | 0x020 | R/W | Peripheral enable register 0 | 0x0 |
| OSMC | 0x023 | R/W | Sub-system clock supply mode control register | 0x0 |

(Base address of the following registers = 0x4002_0810)

RO: Read Only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------|-------------|
| PER11 | 0x001 | R/W | Peripheral enable register 1 | 0x0 |
| PER12 | 0x002 | R/W | Peripheral enable register 1 | 0x0 |
| PER13 | 0x003 | R/W | Peripheral enable register 1 | 0x0 |

(Base address of the following registers = 0x4002_1C00)

RO: Read Only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------------------------------------|--------------------------------------------------------------------|
| HOCODIV | 0x020 | R/W | High-speed on-chip oscillator frequency selection register | The set values of the FRQSEL2~FRQSEL0 bits of option byte (000C2H) |

5.4 Register description

The clock generation circuit is controlled through the following registers.

- System clock control register (CKC)
- Clock operation status control register (CSC)
- Peripheral enable registers 0, 1 (PER0, PER11, PER12, PER13)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Subsystem clock select register (SUBCKSEL)

Note: The assigned registers and bits vary from product to product. The initial values must be set for unassigned bits.

5.4.1 System clock control register (CKC)

This is a register that selects the CPU/peripheral hardware clock and the main system clock.

The CKC register is set by an 8-bit memory manipulation instruction.

| Bit | Symbol | Description | Reset value |
|-----|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | CLS ^{Note 1} | CPU/peripheral hardware clock (F_{CLK}) status 0: Main system clock (F_{MAIN}) 1: Low-speed on-chip oscillator clock (F_{IL}) | 0 |
| 6 | CSS | CPU/ peripheral hardware clock (F_{CLK}) selection 0: Main system clock (F_{MAIN}) 1: Low-speed on-chip oscillator clock (F_{IL}) | 0 |
| 5:0 | -- | Reserved | 0x0 |

Note 1: bit7 is a read-only bit.

Note 2: Bits 0 to 5 must be set to “0”.

Note 3: Clocks set by the CSS bit are provided for the CPU and peripheral hardware. If you change the CPU clock, change the peripheral hardware clock at the same time (except for clock output/buzzer output and watchdog timer). Therefore, if you want to change the clock on the CPU/peripheral hardware, you must stop the peripheral functions.

5.4.2 Clock operation status control register (CSC)

This is a register that controls the operation of the high-speed system clock. The CSC register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "C0H".

| Bit | Symbol | Description | Reset value |
|-----|---------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:1 | -- | Reserved | 0x60 |
| 0 | HIOSTOP | Operation control of high-speed on-chip oscillator clock 0: High-speed on-chip oscillator runs 1: High-speed on-chip oscillator stops | 0 |

Note 1: Do not stop the clock selected for the CPU peripheral hardware clock (F_{CLK}) with the CSC register.

Note 2: For the register flag setting to stop clock oscillation and the conditions before stopping, refer to Table 5-2.

Table 5-2: Condition before stopping clock oscillation

| Clock | Condition before stopping clock | Setting of CSC register flags |
|-------------------------------------|----------------------------------------------------------------------------------------------------------|-------------------------------|
| High-speed on-chip oscillator clock | CPU/peripheral hardware clock runs on a clock other than the high-speed on-chip oscillator clock (CLS=1) | HIOSTOP=1 |

5.4.3 Peripheral enable registers 0, 1 (PER0, PER11, PER12, PER13)

These registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise. When using the following peripheral functions controlled by these registers, the corresponding bit must be set to “1” before initial setting of the peripheral functions.

- LSITIMER 12-bit interval timer
- General-purpose timer unit (TIMER4)
- CCP0/1
- EPWM
- TIMER01
- IIC
- UART
- SPI
- DIVSQRT
- ADCLDO
- DAC
- PGA12
- PGA0
- ACMP0/1
- ADC

The PER0 register and the PER11, PER12, and PER13 registers are set by 8-bit memory manipulation instructions.

After reset signals are generated, the values of these registers change to “00H”.

Peripheral enable register 0 (PER0)

| Bit | Symbol | Description | Reset value |
|-----|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | LSITIMEREN | Control of LSITIMER input clock supply (power-down sleep enabled) 0: Stops input clock supply • SFR used by the LSITIMER cannot be written. • The LSITIMER is in the reset status. 1: Enables input clock supply • SFR used by the LSITIMER can be written. | 0 |
| 6:1 | -- | Reserved | 0x0 |
| 0 | TIM40EN | Control of general-purpose timer unit 4 input clock supply 0: Stops input clock supply • SFR used by the general-purpose timer unit 4 cannot be written. • The general-purpose timer unit 4 is in the reset status. 1: Enables input clock supply • SFR used by the general-purpose timer unit 4 can be written. | 0 |

Peripheral enable register 1 (PER11)

| Bit | Symbol | Description | Reset value |
|-----|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:5 | -- | Reserved | 0x0 |
| 4 | CCPEN | Control of CCP module input clock supply 0: Stops input clock supply • CCP cannot run 1: Enables input clock supply • CCP can run | 0 |
| 3 | EPWMEN | Control of EPWM module input clock supply 0: Stops input clock supply • EPWM cannot run 1: Enables input clock supply • EPWM can run | 0 |
| 2:1 | -- | Reserved | 0x0 |
| 0 | TIMER01EN | Control of TIMER01 input clock supply 0: Stops input clock supply • TIMER01 cannot run 1: Enables input clock supply • TIMER01 can run | 0 |

Peripheral enable register 1 (PER12)

| Bit | Symbol | Description | Reset value |
|-----|--------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:5 | -- | Reserved | 0x0 |
| 4 | IICEN | Control of IIC module input clock supply 0: Stops input clock supply • IIC cannot run 1: Enables input clock supply • IIC can run | 0 |
| 3 | -- | Reserved | 0 |
| 2 | UARTEN | Control of UART module input clock supply 0: Stops input clock supply • UART cannot run 1: Enables input clock supply • UART can run | 0 |
| 1 | SPIEN | Control of SPI module input clock supply 0: Stops input clock supply • SPI cannot run 1: Enables input clock supply • SPI can run | 0 |
| 0 | DIVEN | Control of DIVSQRT input clock supply 0: Stops input clock supply • DIVSQRT cannot run 1: Enables input clock supply • DIVSQRT can run | 0 |

Peripheral enable register 1 (PER13)

| Bit | Symbol | Description | Reset value |
|-----|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | -- | Reserved | 0 |
| 6 | ADCLDOEN | Control of ADCLDO module input clock supply 0: Stops input clock supply • ADCLDO cannot run 1: Enables input clock supply • ADCLDO can run | 0 |
| 5 | DACEN | Control of DAC module input clock supply 0: Stops input clock supply • DAC cannot run 1: Enables input clock supply • DAC can run | |
| 4 | PGA12EN | Control of PGA12 module input clock supply 0: Stops input clock supply • PGA12 cannot run 1: Enables input clock supply • PGA12 can run | 0 |
| 3 | PGA0EN | Control of PGA0 module input clock supply 0: Stops input clock supply • PGA0 cannot run 1: Enables input clock supply • PGA0 can run | 0 |
| 2 | -- | Reserved | 0 |
| 1 | ACMPEN | Control of ACMP0/1 module input clock supply 0: Stops input clock supply • ACMP0/1 cannot run 1: Enables input clock supply • ACMP0/1 can run | 0 |
| 0 | ADCEN | Control of ADC module input clock supply 0: Stops input clock supply • ADC cannot run 1: Enables input clock supply • ADC can run | 0 |

5.4.4 12-bit interval timer operation clock select register (OSMC)

Select the operation clock for the 12-bit interval timer LSITIMER via the OSMC register. The OSMC register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “00H”.

| Bit | Symbol | Description | Reset value |
|-----|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:5 | -- | Reserved | 0x0 |
| 4 | WUTMMCK0 | Selection of operation clock for 12-bit interval timer 0: Stops to supply a clock to the 12-bit interval timer LSITIMER by the low-speed on-chip oscillator clock 1: Supply a clock to the 12-bit interval timer LSITIMER by the low-speed on-chip oscillator clock | 0 |
| 3:0 | -- | Reserved | 0x0 |

5.4.5 High-speed on-chip oscillator frequency select register (HOCODIV)

This is a register that changes the frequency of the high-speed on-chip oscillator set by the option byte (000C2H). However, the frequency that can be selected varies depending on the values of the FRQSEL4 bit and FRQSEL3 bit of the option byte (000C2H).

The HOCODIV register is set by an 8-bit memory manipulation instruction.

After the reset signal is generated, the value of this register changes to the set value of the FRQSEL2 to FRQSEL0 bits of the option byte (000C2H).

| Bit | Symbol | Description | | | Reset value |
|-----|--------------------|----------------------------------------------------------------|----------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------------|
| 7:3 | -- | Reserved | | | 0x0 |
| 2:0 | HOCODIV2~ HOCODIV0 | Selection of clock frequency for high-speed on-chip oscillator | | | Set value of FRQSEL2 to FRQSEL0 bits of option byte (000C2H) |
| | | | FSQSEL4,3=00 | FSQSEL4,3=01 | |
| | | 000 | F _{IH} =48MHZ F _{HOCO} =48MHZ | F _{IH} =64MHZ F _{HOCO} =64MHZ | |
| | | 001 | F _{IH} =24MHZ F _{HOCO} =48MHZ | F _{IH} =32MHZ F _{HOCO} =64MHZ | |
| | | 010 | F _{IH} =12MHZ F _{HOCO} =48MHZ | F _{IH} =16MHZ F _{HOCO} =64MHZ | |
| | | 011 | F _{IH} =6MHZ F _{HOCO} =48MHZ | F _{IH} =8MHZ F _{HOCO} =64MHZ | |
| | | 100 | F _{IH} =3MHZ F _{HOCO} =48MHZ | F _{IH} =4MHZ F _{HOCO} =64MHZ | |
| | | 101 | Settings are prohibited | F _{IH} =2MHZ F _{HOCO} =64MHZ | |
| | | Other than the above | Settings are prohibited | | |

Note 1: The HOCODIV register must be set in the state where the high-speed on-chip oscillator clock (F_{IH}) is selected as the CPU/peripheral hardware clock (F_{CLK}).

Note 2: After changing the frequency via the HOCODIV register, frequency switching is performed after the following transfer times:

- Runs for up to 3 clocks at the frequency before the change.
- Wait for up to 3 CPU/peripheral hardware clocks at the changed frequency.

5.4.6 Low-speed on-chip oscillator clock select register (SUBCKSEL)

The SUBCKSEL register is a register that selects the subsystem clock FSUB and the low-speed on-chip oscillator clock F_{IL} .

The SUBCKSEL register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "00H".

| Bit | Symbol | Description | Reset value |
|-----|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:1 | -- | Reserved | 0x0 |
| 0 | SELLOSC | Low-speed on-chip oscillator clock selection 0: Disable selecting the low-speed on-chip oscillator clock 1: Select the low-speed on-chip oscillator clock | 0 |

5.4.7 Power mode control protection register (PMUKEY)

The PMUKEY register is a register for controlling the protection of PMUCTL by the power supply mode.

The PMUKEY register is set by a 16-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "0000H".

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:0 | PMUKEY | Power mode control protection register selection • Release the PMUCTL write protection. Write control of the PWDNEN bit of PMUCTL is enabled by writing 192AH and 3E4FH to PMUKEY successively. • Other. The PMUCTL write setting is invalid. | 0x0 |

5.4.8 Power mode control register (PMUCTL)

The PMUCTL register is a register that controls the enable power supply control mode.

The PMUCTL register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "00H" and the write protection is enabled, and the write control is released by PMUKEY.

| Bit | Symbol | Description | Reset value |
|-----|--------|-----------------------------------------------------------------------------------------------------------------------|-------------|
| 7:1 | -- | Reserved | 0x0 |
| 0 | PWDNEN | Selection of power mode control register 0: Partial power-down mode disabled 1: Partial power-down mode enabled | 0 |

Note: Release PMUCTL write protection via PMUKEY.

5.5 System clock oscillation circuit

5.5.1 High-speed on-chip oscillator

The CMS32M65xx has a built-in high-speed on-chip oscillator. The frequency can be selected from 64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, and 2MHz using the option byte (000C2H). Oscillation can be controlled by bit0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after power-on reset is released.

5.5.2 Low-speed on-chip oscillator

The CMS32M65xx has a built-in low-speed on-chip oscillator.

The low-speed on-chip oscillator clock is used as the clock for the watchdog timer, the LSITIMER, and the external reference clock for the SysTick timer, as well as the CPU clock and peripheral module clock.

When bit4 (WDTON) of the option byte (000C0H) or bit4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC) is “1”, or when bit0 (SELLOSC) of the sub-system clock selection register (SUBCKSEL) is “1”, the low-speed on-chip oscillator oscillates.

When the watchdog timer stops running and the WUTMMCK0 bit is not “0”, the low-speed on-chip oscillator continues to oscillate. However, if the watchdog timer is running and the WUTMMCK0 bit or the SELLOSC bit is “0”, the low-speed on-chip oscillator stops oscillating when the WDSTBYON bit is “0” and it is in the sleep mode or deep sleep mode. When the watchdog timer is running, the low-speed on-chip oscillator clock does not stop running even if the program is out of control.

5.6 Operation of clock generation circuit

The clock generation circuit generates various clocks as shown below and controls the CPU operation modes such as standby mode (refer to Figure 5-1).

F_{MAIN} : Main system clock frequency

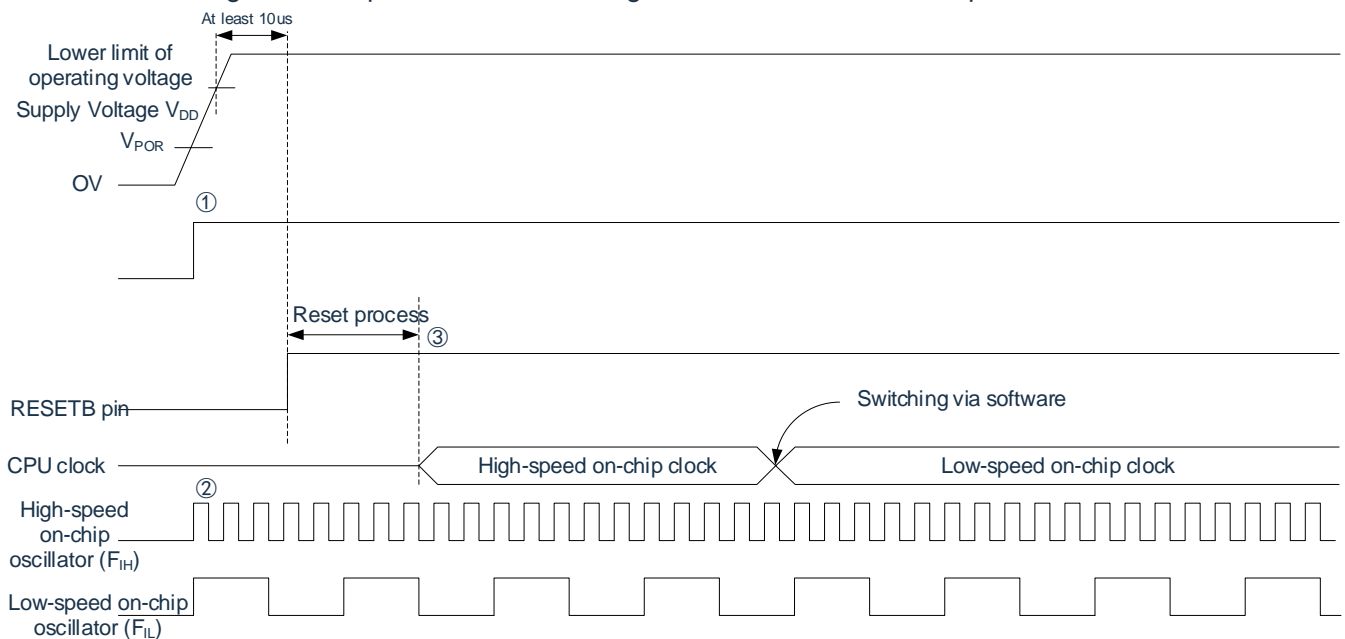
F_{IH} : High-speed on-chip oscillator clock frequency

F_{IL} : Low-speed on-chip oscillator clock frequency

F_{CLK} : CPU/peripheral hardware clock frequency

After the CMS32M65xx is released from reset, the CPU begins operation through the output of the high-speed on-chip oscillator. The operation of the clock generation circuit when the power is turned on is shown in Figure 5-2.

Figure 5-2: Operation of the clock generation circuit when the power is turned on



- 1) After power is turned on, an internal reset signal is generated through the power-on reset (POR) circuit.
- 2) However, the reset state is maintained by a voltage detection circuit or an external reset until the operating voltage range shown in the AC characteristics of the datasheet is reached (the above figure shows an example when an external reset is used).
- 3) The high-speed on-chip oscillator starts oscillating automatically after the reset is released.
- 4) After the reset is released, voltage stabilization waiting and reset processing are performed, and then the CPU starts running with a high-speed on-chip oscillator clock.

5.7 Clock control

5.7.1 Example of setting up a high-speed on-chip oscillator

The CPU/peripheral hardware clock (F_{CLK}) must run at the high-speed on-chip oscillator clock after the reset is released. The frequency of the high-speed on-chip oscillator can be selected from 64MHz, 48MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, and 2MHz by using bits FRQSEL0 to FRQSEL4 of the option byte (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator register (HOCODIV).

[Option byte 000C2 setting]

| Bit | Symbol | Description | | | Reset value |
|-----|------------------|------------------------------------------------------------|--------------------------|-----------------|-------------|
| 7:5 | -- | Reserved | | | -- |
| 4:0 | FRQSEL4~ FRQSEL0 | Selection of high-speed on-chip oscillator clock frequency | | | -- |
| | | | F _{HOCO} | F _{IH} | |
| | | 01000 | 64MHZ | 64MHZ | |
| | | 00000 | 48MHZ | 48MHZ | |
| | | 01001 | 64MHZ | 32MHZ | |
| | | 00001 | 48MHZ | 24MHZ | |
| | | 01010 | 64MHZ | 16MHZ | |
| | | 00010 | 48MHZ | 12MHZ | |
| | | 01011 | 64MHZ | 8MHZ | |
| | | 00011 | 48MHZ | 6MHZ | |
| | | 01100 | 64MHZ | 4MHZ | |
| | | 00100 | 48MHZ | 3MHZ | |
| | | 01101 | 64MHZ | 2MHZ | |
| | | Other than the above | Settings are prohibited. | | |

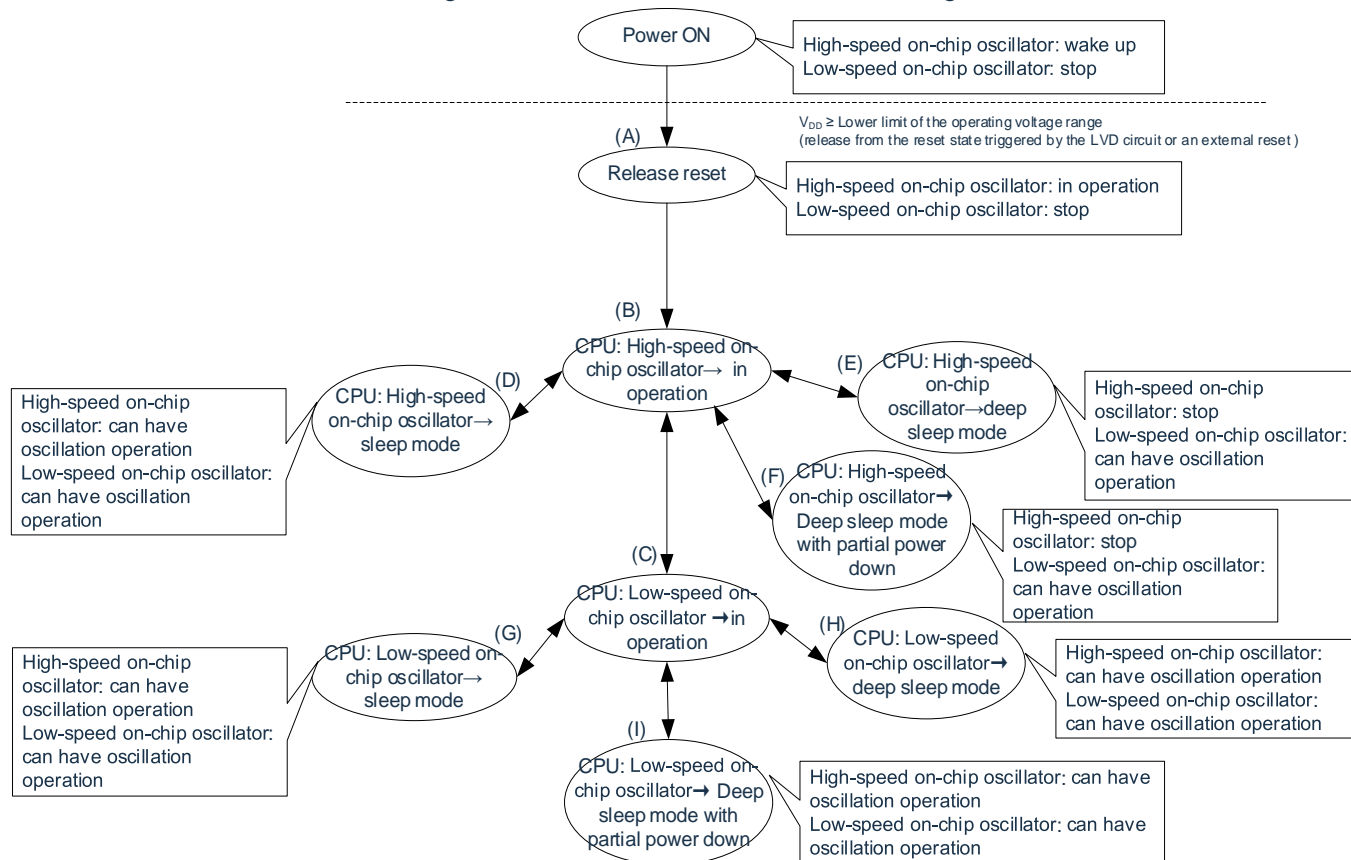
[Setting of high-speed on-chip oscillator frequency select register (HOCODIV)]

| Bit | Symbol | Description | | Reset value |
|-----|-----------------------|------------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------------|
| 7:3 | -- | Reserved | | 0x0 |
| 2:0 | HOCODIV2~ HOCODIV0 | Selection of high-speed on-chip oscillator clock frequency | | Set value of FRQSEL2 to FRQSEL0 bits of option byte (000C2H) |
| | | | FSQSEL4=0 | |
| | | | FSQSEL3=0 FSQSEL3=1 | |
| | | 000 | F _{IH} =48MHZ F _{HOCO} =48MHZ | F _{IH} =64MHZ F _{HOCO} =64MHZ |
| | | 001 | F _{IH} =24MHZ F _{HOCO} =48MHZ | F _{IH} =32MHZ F _{HOCO} =64MHZ |
| | | 010 | F _{IH} =12MHZ F _{HOCO} =48MHZ | F _{IH} =16MHZ F _{HOCO} =64MHZ |
| | | 011 | F _{IH} =6MHZ F _{HOCO} =48MHZ | F _{IH} =8MHZ F _{HOCO} =64MHZ |
| | | 100 | F _{IH} =3MHZ F _{HOCO} =48MHZ | F _{IH} =4MHZ F _{HOCO} =64MHZ |
| | | 101 | Settings are prohibited | F _{IH} =2MHZ F _{HOCO} =64MHZ |
| | | Other than the above | Settings are prohibited | |

5.7.2 CPU clock status transition diagram

Figure 5-3 shows the CPU clock status transition diagram of this product.

Figure 5-3: CPU clock status transition diagram



Examples of CPU clock transition and SFR register setting are in Table 5-3.

Table 5-3: CPU clock transition and SFR register setting examples (1/3)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

| Status transition | SFR register setting |
|-------------------|---------------------------------------------------------------------------|
| (A)→(B) | SFR registers do not have to be set (default status after reset release). |

(2) CPU operating with high-speed system clock (C) after reset release (A) (The CPU operates (B) with a high-speed on-chip oscillator clock immediately after the reset is released)

(SFR register setting order) →

| SFR register setting flag | SUBCKSEL register | CKC register |
|---------------------------|-------------------|--------------|
| | SELLOSC | CSS |
| (B)→(C) | 1 | 1 |

Table 5-3: CPU clock transition and SFR register setting examples (2/3)

(3) The CPU moves from high-speed on-chip clock operation (B) to low-speed on-chip clock operation (C).

(SFR register setting order) →

| State transition | SFR register setting flag | SUBCKSEL register | CKC register |
|------------------|---------------------------|-------------------|--------------|
| | | SELLOSC | CSS |
| (B)→(C) | | 1 | 1 |

(4) The CPU moves from low-speed on-chip clock operation (C) to high-speed on-chip clock operation (B).

(SFR register setting order) →

| State transition | SFR register setting flag | CSC register | Oscillation stabilization waiting | CKC register |
|------------------|---------------------------|--------------|-----------------------------------|--------------|
| | | HIOSTOP | | CSS |
| (C)→(B) | | 0 | 1us | 0 |

Note 1: (A) to (I) of Table 5-3 correspond to (A) to (I) of Figure 5-3.

Note 2: The oscillation accuracy of the high-speed on-chip oscillator clock stabilization wait time varies depending on temperature conditions and during deep sleep mode.

(5) The CPU moves from high-speed on-chip clock operation (B) to sleep mode (D).

The CPU moves from low-speed on-chip clock operation (C) to sleep mode (G).

| Status transition | Setting contents |
|--------------------|------------------------------|
| (B)→(D) (C)→(G) | Execute the WFI instruction. |

Note: (A) to (I) of Table 5-3 correspond to (A) to (I) of Figure 5-3.

(6) The CPU moves from high-speed on-chip clock operation (B) to deep sleep mode (E).

The CPU moves from low-speed on-chip clock operation (C) to deep sleep mode (H).

(Setting order) →

| Status transition | Setting contents | |
|--------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| (B)→(E) (C)→(H) | Stop Peripheral functions that cannot be run in deep sleep mode. | Bit2 of the SCR register (SLEEPDEEP) is set to 1 and the WFI instruction is executed. |

Note: (A) to (I) of Table 5-3 correspond to (A) to (I) of Figure 5-3.

Table 5-3: CPU clock transition and SFR register setting examples (3/3)

(7) The CPU moves from high-speed on-chip clock operation (B) to deep sleep mode with partial power-down (F).

The CPU moves from low-speed on-chip clock operation (C) to deep sleep mode with partial power-down (I).

(Setting order) →

| Status transition | Setting contents | | |
|--------------------|---------------------------------------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------|
| (B)→(E) (C)→(H) | Stop Peripheral functions that cannot be run in deep sleep mode. | PMUKEY=0x192A; PMUKEY=0x3E4F; PMUCTL=0x01; | Bit2 of the SCR register (SLEEPDEEP) is set to 1 and the WFI instruction is executed. |

Note: (A) to (I) of Table 5-3 correspond to (A) to (I) of Figure 5-3.

5.7.3 Conditions before CPU clock transfer and post-transfer processing

The conditions before the CPU clock transfer and the processing after the transfer are shown below.

Table 5-4: Transfer of CPU clocks

| CPU clock | | Conditions before transfer | Post-transfer processing |
|-------------------------------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before transfer | After transfer | | |
| High-speed on-chip oscillator clock | Low-speed on-chip oscillator clock | Selects low-speed on-chip oscillator clock SELLOSC =1 | If the oscillation of the high-speed on-chip oscillator is stopped (HIOSTOP=1), the operation current can be reduced. |
| Low-speed on-chip oscillator clock | High-speed on-chip oscillator clock | Enables high-speed on-chip oscillator to oscillate. • HIOSTOP=0 • After oscillation stabilization time | If the watchdog and LSITIMER are not running, the operating current can be reduced by turning off the low-speed on-chip clock oscillation (SELLOSC =0). |

5.7.4 Time required to switch CPU clock and main system clock

It can switch CPU clock (main system clock↔sub system clock) and main system clock (high speed on-chip oscillator clock↔high speed system clock) by setting bit6 (CSS) of system clock control register.

The actual switchover does not occur immediately after the CKC register is overridden, but several clocks continue to run with the clock before the switchover after the CKC register is changed (see Table 5-5).

The CPU can be judged by the bit7 (CLS) of the CKC register whether the CPU is run with the main system clock or the low-speed on-chip oscillator clock.

If you switch the CPU clock, switch the peripheral hardware clock at the same time.

Table 5-5: Maximum number of clocks required for $F_{MAIN} \leftrightarrow F_{IL}$

| Set value before switching | Set value after switching | |
|-----------------------------|-----------------------------|------------------------------|
| CSS | CLS | |
| | 0 ($F_{CLK}=F_{MAIN}$) | 1 ($F_{CLK}=F_{IL}$) |
| 0 ($F_{CLK}=F_{MAIN}$) | 2 clocks | 2.5 F_{MAIN}/F_{IL} clocks |
| 1 ($F_{CLK}=F_{IL}$) | | |

Note 1: The number of clocks in Table 5-5 is the number of CPU clocks before the switch.

Note 2: The number of clocks Table 5-5 is the number of clocks rounded to the decimal portion.

Example: When switching CPU from the main system clock to sub-system clock (oscillation with $F_{IH}=2\text{MHz}$, $F_{SUB}=F_{IL}=15\text{KHz}$)

$$2.5F_{MAIN}/F_{SUB}=2.5(2000/15)=333.3 \rightarrow 334 \text{ clocks}$$

5.7.5 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation and conditions before the clock oscillation is stopped.

Table 5-6: Conditions and flag settings before clock oscillation stops

| Clock | Conditions before clock oscillation is stopped | Flag settings of SFR register |
|-------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------|
| High-speed on-chip oscillator clock | CLS=1 (CPU runs at a clock other than the high-speed on-chip oscillator clock) | HIOSTOP=1 |
| Low-speed on-chip oscillator clock | CLS=0, WDTON=0, WUMMCK0=0 (CPU runs at a clock other than the low-speed on-chip oscillator clock) | SELLOSC =0 |

Chapter 6 General-Purpose Timer Unit

6.1 Function of general-purpose timer unit

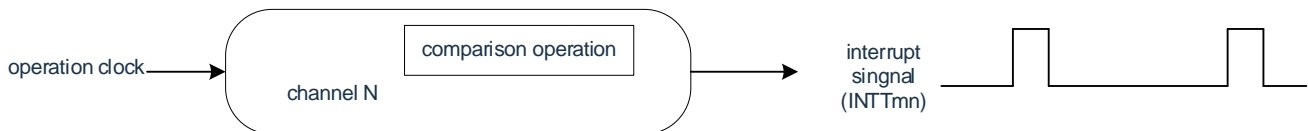
The general-purpose timer unit has the following functions:

6.1.1 Independent channel operation function

The independent channel operation function is a function that enables independent use of any channel without being affected by other channel operation modes.

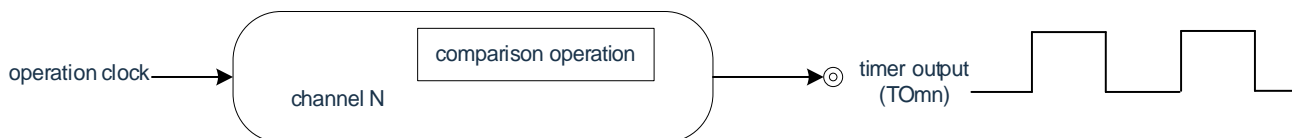
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



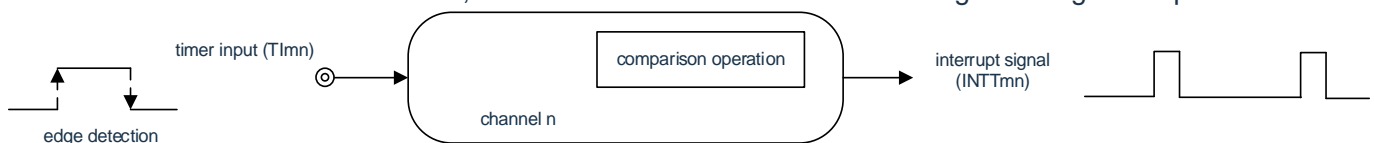
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty cycle of 50% is output from a timer output pin (TOMn).



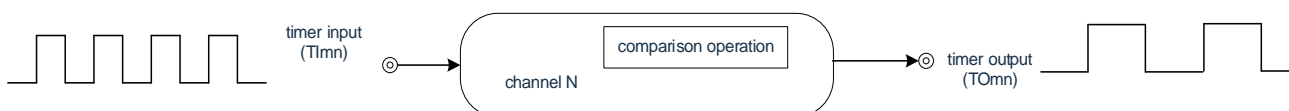
(3) External event counter

The valid edge of the input signal of the timer input pin (TIMn) is counted, and if the specified number of times is reached, it can be used as an event counter for generating interrupts.



(4) Frequency divider function (channel 0 of unit 0 only)

The input clock from the timer input pin (TI00) is divided and then output from the output pin (TO00).



(5) Measurement of input pulse interval

The interval between input pulses is measured by starting counting at the active edge of the input pulse signal at the timer input pin (TIMn) and capturing the count value at the active edge of the next pulse.



(6) Measurement of the high-/low-level width of the input signal

The high- and low-level width of the input signal is measured by starting the count on one edge of the input signal at the timer input pin (TImn) and capturing the count value on the other edge.



(7) Delay counter

Counting begins on the active edge of the input signal to the timer input pin (TImn) and an interrupt is generated after an arbitrary delay period.



Note 1: m: unit number (m=0) n: channel number (n=0~3)

Note 2: Please refer to Chapter 3 Pin Function for the configurable timer input/output pins of channel 0~3.

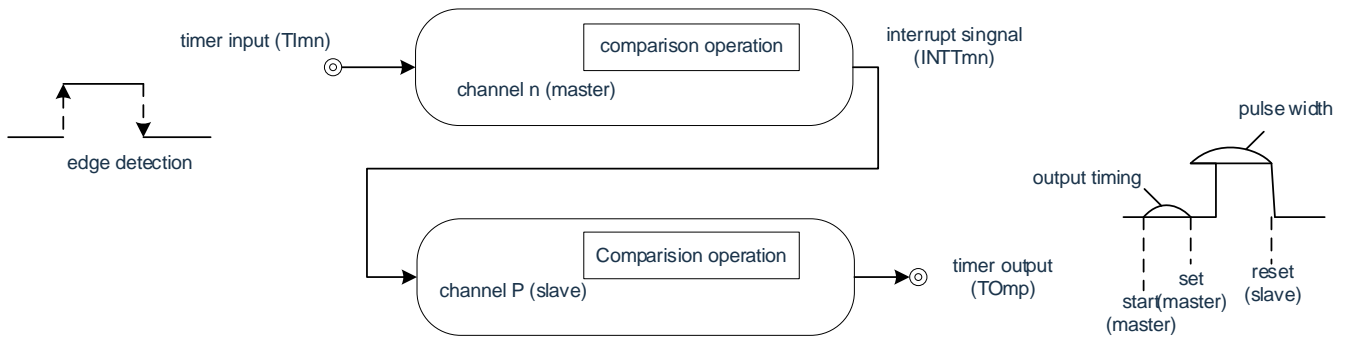
6.1.2 Multi-channel linkage operation functions

The multi-channel linked operation function is a combination of a master channel (the reference timer for the master control cycle) and a slave channel (a timer that operates in compliance with the master channel).

The multi-channel linkage operation function can be used as the following modes.

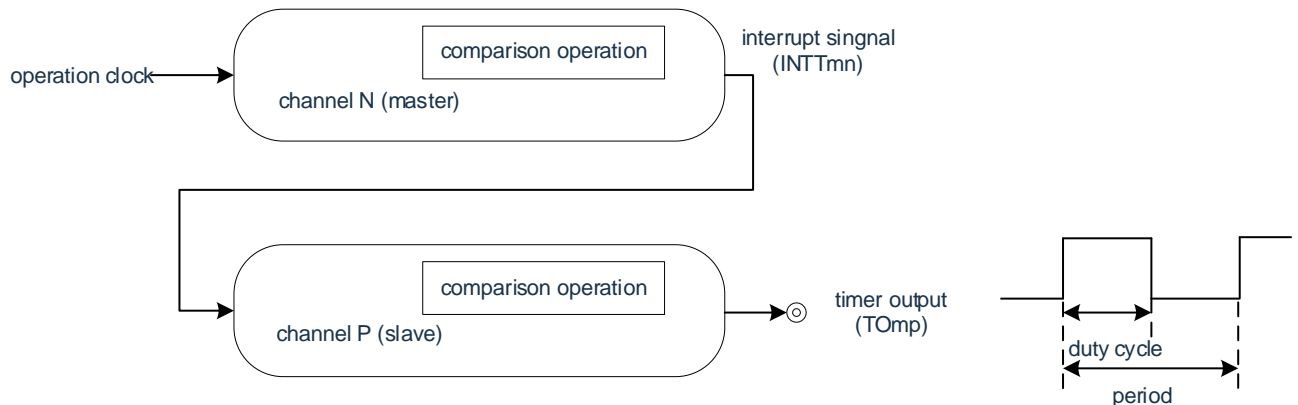
(1) Single trigger pulse output

Using the 2 channels in pairs, a single trigger pulse with arbitrary output timing and pulse width can be generated.



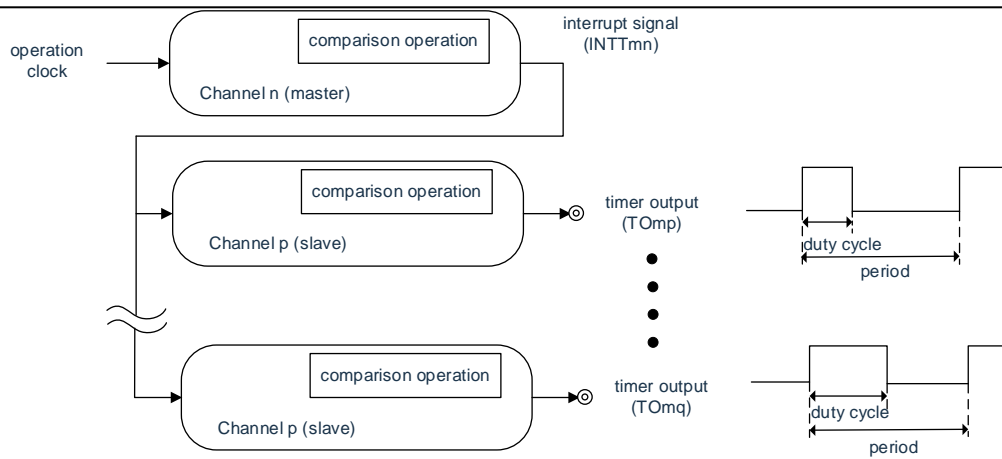
(2) PWM (Pulse Width Modulation) output

Using the 2 channels in pairs, pulses with arbitrary period and duty cycle can be generated.



(3) Multiple PWM (Pulse Width Modulation) output

The PWM function can be extended to generate up to 3 PWM signals of any duty cycle with a fixed period using one master channel and multiple slave channels.



Note 1: Please refer to “6.3.1 Basic rules of multi-channel linkage operation function” for the rule details of multi-channel linkage operation function.

Note 2: m: unit number (m=0) n: channel number (n=0~3) p, q: slave channel number (n<p<q≤3)

6.1.3 8-bit timer operation function (channels 1 and 3 of unit 0 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Note: There are several rules for using 8-bit timer operation function.

For details, see 6.3.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

6.2 Structure of general-purpose timer unit

The general-purpose timer unit consists of the following hardware.

Table 6-1: Structure of general-purpose timer unit

| Item | Structure |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Counter | Timer count register mn (TCRmn) |
| Register | Timer data register mn (TDRmn) |
| Timer input | TI00~TI03 ^{Note 1} |
| Timer output | TO00~TO03 ^{Note 1} , output control circuit |
| Control registers | <Registers of unit setting section> <ul style="list-style-type: none"> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input/output select register 0 (TIOS0)^{Note 2} • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm) |
| | <Registers of each channel> <ul style="list-style-type: none"> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable register 1, 2 (NFEN1, NFEN2) • Port mode control register (PMCxx)^{Note 3} • Port mode register (PMxx)^{Note 3} • Port output multiplexing function configuration register (PxxCFG)^{Note 3} • Port input multiplexing function configuration register (TI1XPCFG)^{Note 3} |

Note 1: The input/output pins of general-purpose timer unit 0 are multiplexed to fixed ports. For details, refer to “Chapter 3 Pin Function”.

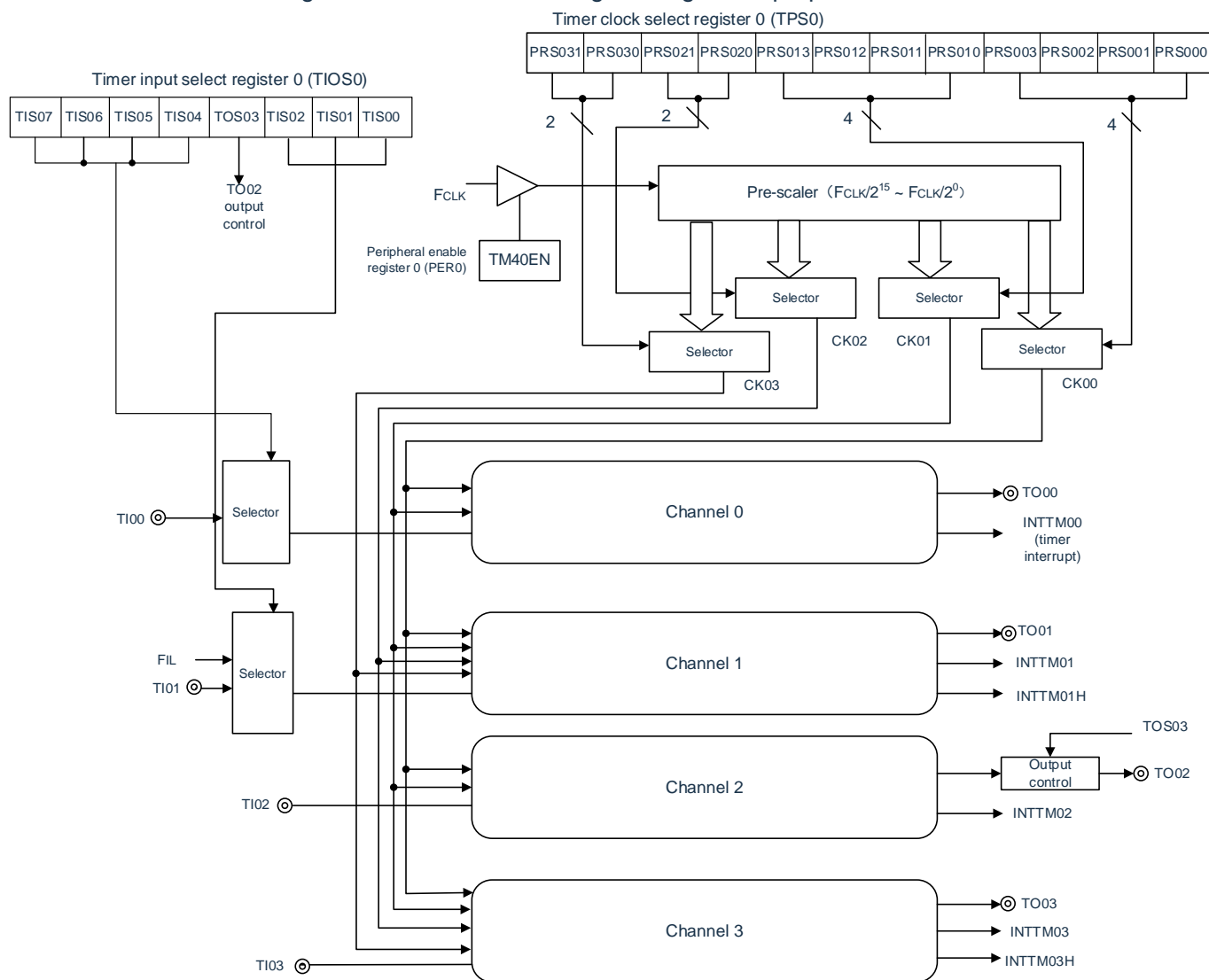
Note 2: Only for channel selection of unit 0.

Note 3: Timer input/output pin configuration for channel 0~3. For details, please refer to “Chapter 3 Pin Function”.

Note 4: m: unit number (m=0) n: channel number (n=0~3)

The block diagram of the general-purpose timer unit is shown in Figure 6-1.

Figure 6-1: Overall block diagram of general-purpose timer unit 0



6.2.1 Register mapping

(Base address of the following registers = 0x4004_1D80)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register name | Offset address | R/W | Bit width | Description | Reset value |
|---------------|----------------|-----|-----------|---------------------------------------|-------------|
| TCR00 | 0x000 | R | 16 | Timer channel 0 count register | FFFFH |
| TCR01 | 0x002 | R | 16 | Timer channel 1 count register | FFFFH |
| TCR02 | 0x004 | R | 16 | Timer channel 2 count register | FFFFH |
| TCR03 | 0x006 | R | 16 | Timer channel 3 count register | FFFFH |
| TMR00 | 0x010 | R/W | 16 | Timer channel 0 mode register | 0000H |
| TMR01 | 0x012 | R/W | 16 | Timer channel 1 mode register | 0000H |
| TMR02 | 0x014 | R/W | 16 | Timer channel 2 mode register | 0000H |
| TMR03 | 0x016 | R/W | 16 | Timer channel 3 mode register | 0000H |
| TSR00 | 0x020 | R | 16 | Timer channel 0 status register | 0000H |
| TSR01 | 0x022 | R | 16 | Timer channel 1 status register | 0000H |
| TSR02 | 0x024 | R | 16 | Timer channel 2 status register | 0000H |
| TSR03 | 0x026 | R | 16 | Timer channel 3 status register | 0000H |
| TE0 | 0x030 | R | 16 | Timer channel enable status register | 0000H |
| TS0 | 0x032 | R/W | 16 | Timer channel start register | 0000H |
| TT0 | 0x034 | R/W | 16 | Timer channel stop register | 0000H |
| TPS0 | 0x036 | R/W | 16 | Timer clock select register | 0000H |
| TO0 | 0x038 | R/W | 16 | Timer output register | 0000H |
| TOE0 | 0x03A | R/W | 16 | Timer output enable register | 0000H |
| TOL0 | 0x03C | R/W | 16 | Timer output level register | 0000H |
| TOM0 | 0x03E | R/W | 16 | Timer output mode register | 0000H |
| TOM0L | 0x03E | R/W | 8 | Timer output mode register low 8 bits | 00H |
| TDR00 | 0x198 | R/W | 16 | Timer channel 0 data register | 0000H |
| TDR01 | 0x19A | R/W | 16 | Timer channel 1 data register | 0000H |
| TDR02 | 0x1E4 | R/W | 16 | Timer channel 2 data register | 0000H |
| TDR03 | 0x1E6 | R/W | 16 | Timer channel 3 data register | 0000H |

(Base address of the following registers = 0x4004_0470)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register name | Offset address | R/W | Bit width | Description |
|---------------|----------------|-----|-----------|------------------------------------|
| NFEN1 | 0x001 | R/W | | Noise filter enable register 1 |
| TIOS0 | 0x004 | R/W | | Timer input/output select register |

6.2.2 Timer count register mn (TCRmn)

The TCRmn register is a 16-bit read-only register that counts the count clock. The count is incremented or decremented synchronously with the rising edge of the count clock.

The operation mode is selected by the MDmn3 to MDmn0 bits of the Timer Mode Register mn (TMRmn) to switch between incremental and decremental counting (refer to “6.2.6: Timer Mode Register mn (TMRmn)”).

Table 6-2: Table of timer count register mn (TCRmn)

| Bit | Symbol | Description | Resr value |
|------|--------|----------------------------------|------------|
| 15:0 | TCRmn | Timer count register (read-only) | 0xFFFF |

Note: m: unit number (m=0) n: channel number (n=0~3)

The count value can be read by reading the timer count register mn (TCRmn).

In the following cases, the count value becomes “FFFFH”.

- When a reset signal is generated
- When clearing the TM4mEN bit of the peripheral enable register 0 (PER0)
- At the end of the count of the slave channel in PWM output mode
- At the end of the count of the slave channel in delayed count mode
- At the end of counting of master/slave channels in single trigger pulse output mode
- At the end of the count of the slave channel in the multiple PWM output mode

In the following cases, the count value becomes “0000H”.

- When input starts triggering in capture mode
- At the end of the capture in capture mode

Note: Even if the TCRmn register is read, the count value is not captured to the timer data register mn (TDRmn).

As shown below, the read values of the TCRmn register vary depending on the operating mode and operating state.

Table 6-3: The read value of the Timer Count Register mn (TCRmn) in various operating mode s

| Operation mode | Counting method | Timer Count Register mn (TCRmn) read value ^{Note1} | | | |
|-----------------------------|-----------------|---------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------|
| | | Value if the operation mode was changed after releasing reset | Counting pause Value at (TTmn = 1) | Counting pause (TTmn=1) after changing the value of the operating mode | Wait after a single count The value at the start of the trigger |
| Interval timer mode | Count down | FFFFH | value when stopped | undefined | - |
| Capture Mode | Count up | 0000H | value when stopped | undefined | - |
| Event counter mode | Count down | FFFFH | value when stopped | undefined | - |
| Single count mode | Count down | FFFFH | value when stopped | undefined | FFFFH |
| Capture & Single Count Mode | Count up | 0000H | value when stopped | undefined | TDRmn register capture value +1 |

Note 1: It indicates the read value of the TCRmn register when channel n is in the timer stop state (TEmn=0) and the count enable state (Tsmn=1). Hold this value in the TCRmn register until counting starts.

Note 2: m: unit number (m=0) n: channel number (n=0~3)

6.2.3 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected. The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to rewrite the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to “0000H”.

Table 6-4: Channel 0 timer data register TDR00

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------|-------------|
| 15:0 | TDR00 | Timer channel 0 data register | 0x0 |

Table 6-5: Channel 1 timer data register TDR01

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------|-------------|
| 15:8 | TDR01H | Timer channel 1 data register bit15:8 | 0x0 |
| 7:0 | TDR01L | Timer channel 1 data register bit7:0 | 0x0 |

Table 6-6: Channel 2 timer data register TDR02

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------|-------------|
| 15:0 | TDR02 | Timer channel 2 data register | 0x0 |

Table 6-7: Channel 3 timer data register TDR03

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------|-------------|
| 15:8 | TDR03H | Timer channel 3 data register bit15:8 | 0x0 |
| 7:0 | TDR03L | Timer channel 3 data register bit7:0 | 0x0 |

(1) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Note: Even if a capture trigger signal is input, the TDRmn register set to the compare function does not perform capture operation.

(2) When timer data register mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Note: m: unit number (m=0) n: channel number (n=0~3)

6.2.4 Peripheral enable register 0 (PER0)

The PER0 register is a register that sets whether to enable or disable the supply of clocks to each peripheral hardware. Reduce power consumption and noise by stopping clocks to hardware that is not in use.

To use general-purpose timer unit 4, bit0 (TM40EN) must be set to “1”. The PER0 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the PER0 register changes to “00H”.

Table 6-8: Table of peripheral enable register 0 (PER0)

| Bit | Symbol | Description | Reset value |
|-----|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | LSITIMEEN | Control of the input clock of 12-bit interval timer 0: Stops input clock supply. • The SFR used by 12-bit interval timer. • 12-bit interval timer is in the reset state. 1: Enables input clock supply. • The SFR used by 12-bit interval timer. | 0 |
| 6:0 | - | Reserved | 0x0 |
| 0 | TM40EN | Control of the input clock of general-purpose timer unit 4 0: Stops input clock supply. • The SFR used by general-purpose timer unit cannot be written. • General-purpose timer unit 4 is in the reset state. 1: Enables input clock supply. • The SFR used by general-purpose timer unit 4 can be read and written. | 0 |

Note: To set the general-purpose timer unit, the following registers must be set with the TM4mEN bit at “1”. When the TM4mEN bit is “0”, the values of the Timer Array Unit’s control registers are initialized, and write operations are ignored (timer input/output select register 0 (TIOS0), noise filter enable register 1 (NFEN1), noise filter enable register 2 (NFEN2), port mode control register (PMCx), port mode register (PMx), and port multiplexing function configuration register (PxxCFG) are excluded).

- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)

6.2.5 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that selects the two or four common operating clocks (CKm0, CKm1, CKm2, CKm3) provided to each channel. CKm0 is selected via bits 3~0 of the TPSm register, and CKm1 is selected via bits 7~4 of the TPSm register. In addition, only channel 1 and channel 3 can select CKm2 and CKm3, and CKm2 is selected via bits 9~8 of the TPSm register, and CKm3 is selected via bits 13 and 12 of the TPSm register.

The TPSm register in timer operation can only be rewritten in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 3):

All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):

All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):

All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).

If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):

All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of the TPSm register changes to "0000H".

Table 6-9: Table of timer clock select register m (TPSm)

| Bit | Symbol | Description | Reset value |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:14 | - | Must be set to 0 | 0x0 |
| 13:12 | CKm3 | Timer operation clock selection CKm3: 00H: Fclk=Fclk/2 ⁸ 01H: Fclk=Fclk/2 ¹⁰ 02H: Fclk=Fclk/2 ¹² 03H: Fclk=Fclk/2 ¹⁴ | 0x0 |
| 11:10 | - | Must be set to 0 | 0x0 |
| 9:8 | CKm2 | Timer operation clock selection CKm2: 00H: Fclk=Fclk/2 01H: Fclk=Fclk/2 ² 02H: Fclk=Fclk/2 ⁴ 03H: Fclk=Fclk/2 ⁶ | 0x0 |
| 7:4 | CKm1 | Timer operation clock selection Fclk=Fclk/2 ^{CKm1} | 0x0 |
| 3:0 | CKm0 | Timer operation clock selection Fclk=Fclk/2 ^{CKm0} | 0x0 |

Note 1: In case of changing the clock selected as F_{CLK} (changing the value of the system clock control register (CKC)), the general-purpose timer unit must be stopped (TTm=0,100FH). The general-

purpose timer unit needs to be stopped even when the operation clock (F_{MCK}) is selected or when the active edge of the TImn pin input signal is used.

Note 2: F_{CLK} : CPU/peripheral hardware clock frequency

Note 3: The clock waveform selected by the TPSm register is high for only 1 F_{CLK} cycle from the rising edge. For details, refer to “6.4.1 Counting Clock (F_{TCLK})”.

Note 4: Bits 15, 14, 11 and 10 must be set to “0”.

Note 5: If F_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to “0000H” (n=0, 1, m=0~3), the interrupt request of general-purpose timer unit cannot be used.

If channels 1 and 3 are used in 8-bit timer mode and CKm2 and CKm3 are used as the operation clocks, the interval times shown in the table below can be realized with the interval timer function.

Table 6-10: Interval time that can be set by operation clocks CKSm2 and CKSm3

| Clock | | Interval time ^{Note} ($F_{CLK}=32\text{MHz}$) | | | |
|-------|------------------|----------------------------------------------------------|-------|-----|------|
| | | 10us | 100us | 1ms | 10ms |
| CKm2 | $F_{CLK}/2$ | ○ | — | — | — |
| | $F_{CLK}/2^2$ | ○ | — | — | — |
| | $F_{CLK}/2^4$ | ○ | ○ | — | — |
| | $F_{CLK}/2^6$ | ○ | ○ | — | — |
| CKm3 | $F_{CLK}/2^8$ | — | ○ | ○ | — |
| | $F_{CLK}/2^{10}$ | — | ○ | ○ | — |
| | $F_{CLK}/2^{12}$ | — | — | ○ | ○ |
| | $F_{CLK}/2^{14}$ | — | — | ○ | ○ |

Note 1: ○The margin is within 5%.

Note 2: F_{CLK} : CPU/peripheral hardware clock frequency

Note 3: For details about the $F_{CLK}/2^n$ waveform selected for the TPSm register, refer to “6.4.1 Count Clock (F_{TCLK})”.

6.2.6 Timer mode register mn (TMRmn)

The TMRmn register is a register that sets the operation mode of channel n. It performs the selection of the operation clock (F_{MCK}), the selection of the count clock, the selection of master/slave, the selection of the 16-bit/8-bit timer (limited to channel 1 and channel 3), the setting of the start trigger and the capture trigger, the selection of the effective edge of the timer input, and the operation modes (interval, capture, event counter, single count, capture & single count) settings.

It is prohibited to rewrite the TMRmn register during operation ($TE_{mn}=1$). However, bit7 and bit6 (CIS_{mn1} , CIS_{mn0}) can be rewritten during part of the function operation ($TE_{mn}=1$) (for details, refer to “6.7 Independent Channel Operation Function of General-Purpose Timer Unit” and “6.8 Multi-Channel Operation Function of General-Purpose Timer Unit”).

The TMRmn register is set by a 16-bit memory manipulation instruction. After a reset signal is generated, the value of the TMRmn register changes to “0000H”.

Caution: Bit11 of the TMRmn register varies from channel to channel.

TMRm: MASTERmn bit ($n=2$)

TMRm1, TMRm3: SPLITmn bit ($n=1, 3$)

TMRm0: Fixed to “0”.

Table 6-11: Timer channel 0 mode register TMR00

| Bit | Symbol | Description | Reset value |
|-------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:14 | CKS001- CKS000 | Selection of channel n operation clock (F_{MCK}) CKS001- CKS000: 00H: The operation clock CKm0 set by the timer clock select register m (TPSm) 01H: The operation clock CKm2 set by the timer clock select register m (TPSm) 02H: The operation clock CKm1 set by the timer clock select register m (TPSm) 03H: The operation clock CKm3 set by the timer clock select register m (TPSm) The operation clock (F_{MCK}) is used for edge detection circuits. The sample clock and count clock (F_{TCLK}) are generated by setting the CCSmn bit. Only Channel 1 and Channel 3 can select operation clocks CKm2 and CKm3. | 0x0 |
| 13 | 0 | Must be set to 0 | 0 |
| 12 | CCS00 | Selection of channel n count clock (F_{TCLK}) 00H: CKS000 bit and CKS001 bit specified operation clock (F_{MCK}) 01H: Channel 0: The active edge of the input signal selected by TIOS0 Channel 1: The active edge of the input signal selected by TIOS0 Counting clocks (F_{TCLK}) are used in counters, output control circuits, and interrupt control circuits. | 0 |

| 11 | 0 | Must be set to 0 | 0 |
|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 10:8 | STS002- STS000 | <p>Start trigger and capture trigger settings for channel n</p> <p>STS002-STS000:</p> <p>00H: Only software triggering is active at the start (no other trigger source is selected).</p> <p>01H: Use the active edge of the TI00 pin input for start triggering and capture triggering.</p> <p>02H: Use the double edges of the TI00 pin input for start triggering and capture triggering respectively.</p> <p>04H: Use interrupt signals from the master channel (in the case of slave channels with multi-channel linkage operation function).</p> <p>Other than the above, settings are prohibited.</p> | 0x0 |
| 7:6 | CIS001- CIS000 | <p>Active edge selection for TI00 pins</p> <p>CIS001-CIS000:</p> <p>00H: Falling edge</p> <p>01H: Rising edge</p> <p>02H: Double edge (when measuring low level width)</p> <p>Start trigger: falling edge, capture trigger: rising edge</p> <p>03H: Double edge (when measuring high level width)</p> <p>Start trigger: rising edge, capture trigger: falling edge</p> | 0x0 |
| 5:4 | 0 | Reserved | 0x0 |
| 3:0 | MD003- MD000 | <p>Setting of channel n operation mode and interrupt</p> <p>MD003-MD000:</p> <p>00H: Interval timer mode, no timer interrupt is generated at the start of counting.</p> <p>01H: Interval timer mode, a timer interrupt is generated when counting starts.</p> <p>04H: Capture mode, no timer interrupt is generated when counting starts.</p> <p>05H: Capture mode, no timer interrupt is generated when counting starts.</p> <p>06H: Event counter mode, no timer interrupt is generated when counting starts.</p> <p>08H: Single count mode, the start trigger in the count operation is invalid. No interruption at this time.</p> <p>09H: Single count mode, the start trigger in the count operation is valid. No interruption at this time.</p> <p>0CH: Capture & single count mode, no timer interrupt is generated when counting starts.</p> | |

For a detailed description of MD003- MD000, see the following table

| MD 003 | MD 002 | MD 001 | Setting of channel n operation mode | Corresponding functions | Count operation of TCR |
|---------------------------------------------------------------------------------|--------|--------|-------------------------------------|--------------------------------------------------------------------------------------|------------------------|
| 0 | 0 | 0 | Interval timer mode | Interval timer/square wave output/ Frequency divider function/PWM output (master) | Count down |
| 0 | 1 | 0 | Capture mode | Measurement of input pulse interval | Count up |
| 0 | 1 | 1 | Event counter mode | External event counter | Count down |
| 1 | 0 | 0 | Single count mode | Delay counter/single trigger pulse output/PWM output (slave) | Count down |
| 1 | 1 | 0 | Capture & Single count mode | Measurement of the high- and low-level width of the input signal | Count up |
| Other than the above | | | Settings are prohibited. | | |
| The operation of each mode varies depending on MD000 bit (see the table below). | | | | | |

| Operation mode (Value set by the MD003 to MD001 bits (see table above)) | MD 000 | Setting of starting counting and interrupt |
|-----------------------------------------------------------------------------------------------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Interval timer mode (0, 0, 0) Capture mode (0, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| | 1 | A timer interrupt is generated when counting starts (the output of the timer also changes). |
| <ul style="list-style-type: none"> Event counter mode (0, 1, 1) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| <ul style="list-style-type: none"> Single count mode ^{Note 1} (1, 0, 0) | 0 | The start trigger in the count operation is invalid. No interruption at this time. |
| | 1 | The start trigger in the count operation is valid ^{Note 2} . No interruption at this time. |
| <ul style="list-style-type: none"> Capture & single count mode (1, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). The start trigger in the count operation is invalid. No interruption at this time. |

Note 1: In single count mode, the interrupt output (INTTM00) and TO00 output at the start of counting are not controlled.

Note 2: If a start trigger is generated during operation (TS00=1), the counter is initialized and counting is restarted (no interrupt request is generated).

Table 6-12: Timer channel 0 mode register TMR01

| Bit | Symbol | Description | Reset value |
|-------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:14 | CKS011- CKS010 | Selection of channel n operation clock (F_{MCK}) CKS011- CKS010: 00H: The operation clock CKm0 set by the timer clock select register m (TPSm). 01H: The operation clock CKm2 set by the timer clock select register m (TPSm). 02H: The operation clock CKm1 set by the timer clock select register m (TPSm). 03H: The operation clock CKm3 set by the timer clock select register m (TPSm). The operation clock (F_{MCK}) is used for edge detection circuits. The sample clock and count clock (F_{TCLK}) are generated by setting the CCSmn bit. Only Channel 1 and Channel 3 can select operation clocks CKm2 and CKm3. | 0x0 |
| 13 | 0 | Must be set to 0 | 0 |
| 12 | CCS01 | Selection of channel n count clock (F_{TCLK}) 00H: CKS010 bit and CKS011 bit specified operation clock (F_{MCK}) 01H: The active edge of the TI01 pin input signal Channel 0: The active edge of the input signal selected by TIOS0 Channel 1: The active edge of the input signal selected by TIOS0 Counting clocks (F_{TCLK}) are used in counters, output control circuits, and interrupt control circuits. | 0 |
| 11 | SPLIT0 | Operation selection of 8-bit timer/16-bit timer for channel 1 00H: Used as a 16-bit timer. (Used as a slave channel for independent channel operation or multi-channel linkage operation) 01H: Used as an 8-bit timer. | 0 |
| 10:8 | STS012- STS010 | Start trigger and capture trigger settings for channel n STS012- STS010: 00H: Only software triggering is active at the start (no other trigger source is selected). 01H: Use the active edge of the TI01 pin input for start triggering and capture triggering. 02H: Use the double edges of the TI01 pin input for start triggering and capture triggering respectively. 04H: Use interrupt signals from the master channel (in the case of slave channels with multi-channel linkage operation function). Other than the above, settings are prohibited. | 0x0 |
| 7:6 | CIS011- CIS010 | Active edge selection for TI01 pins CIS011- CIS010 00H: Falling edge 01H: Rising edge 02H: Double edge (when measuring low-level | 0x0 |

| | | | |
|-----|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | width) Start trigger: falling edge, capture trigger: rising edge 03H: Double edge (when measuring high-level width) Start trigger: rising edge, capture trigger: falling edge | |
| 5:4 | 0 | Reserved | 0x0 |
| 3:0 | MD013- MD010 | Setting of channel n operation mode and interrupt MD013- MD010: Setting of channel n operation mode and interrupt MD013- MD010: 00H: Interval timer mode, no timer interrupt is generated at the start of counting. 01H: Interval timer mode, a timer interrupt is generated when counting starts. 04H: Capture mode, no timer interrupt is generated when counting starts. 05H: Capture mode, no timer interrupt is generated when counting starts. 06H: Event counter mode, no timer interrupt is generated when counting starts. 08H: Single count mode, the start trigger in the count operation is invalid. No interruption at this time. 09H: Single count mode, the start trigger in the count operation is valid. No interruption at this time. 0CH: Capture & single count mode, no timer interrupt is generated when counting starts. | |

For a detailed description of MD013- MD010, see the following table

| MD 013 | MD 012 | MD 011 | Setting of channel n operation mode | Corresponding functions | Count operation of TCR |
|---------------------------------------------------------------------------------|--------|--------|-------------------------------------|--------------------------------------------------------------------------------------|------------------------|
| 0 | 0 | 0 | Interval timer mode | Interval timer/square wave output/ Frequency divider function/PWM output (master) | Count down |
| 0 | 1 | 0 | Capture mode | Measurement of input pulse interval | Count up |
| 0 | 1 | 1 | Event counter mode | External event counter | Count down |
| 1 | 0 | 0 | Single count mode | Delay counter/single trigger pulse output/PWM output (slave) | Count down |
| 1 | 1 | 0 | Capture & Single count mode | Measurement of the high- and low-level width of the input signal | Count up |
| Other than the above | | | Settings are prohibited. | | |
| The operation of each mode varies depending on MD010 bit (see the table below). | | | | | |

| | | |
|-------------------------------------------------------------------------------|--------|--------------------------------------------|
| Operation mode (Value set by the MD013 to MD011 bits (see table above)) | MD 010 | Setting of starting counting and interrupt |
|-------------------------------------------------------------------------------|--------|--------------------------------------------|

| | | |
|-----------------------------------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Interval timer mode (0, 0, 0) Capture mode (0, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| | 1 | A timer interrupt is generated when counting starts (the output of the timer also changes). |
| <ul style="list-style-type: none"> Event counter mode (0, 1, 1) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| <ul style="list-style-type: none"> Single count mode ^{Note 1} (1, 0, 0) | 0 | The start trigger in the count operation is invalid. No interruption at this time. |
| | 1 | The start trigger in the count operation is valid ^{Note 2} . No interruption at this time. |
| <ul style="list-style-type: none"> Capture & single count mode (1, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). The start trigger in the count operation is invalid. No interruption at this time. |

Note 1: In single count mode, the interrupt output (INTTM01) and TO01 output at the start of counting are not controlled.

Note 2: If a start trigger is generated during operation (TS01=1), the counter is initialized and counting is restarted (no interrupt request is generated).

Table 6-13: Timer channel 1 mode register TMR02

| Bit | Symbol | Description | Reset value |
|-------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:14 | CKS021- CKS020 | Selection of channel n operation clock (F_{MCK}) CKS021- CKS020: 00H: The operation clock CKm0 set by the timer clock select register m (TPSm) 01H: The operation clock CKm2 set by the timer clock select register m (TPSm) 02H: The operation clock CKm1 set by the timer clock select register m (TPSm) 03H: The operation clock CKm3 set by the timer clock select register m (TPSm) The operation clock (F_{MCK}) is used for edge detection circuits. The sample clock and count clock (F_{TCLK}) are generated by setting the CCSmn bit. Only Channel 1 and Channel 3 can select operation clocks CKm2 and CKm3. | 0x0 |
| 13 | 0 | Must be set to 0 | 0 |
| 12 | CCS02 | Selection of channel n count clock (F_{TCLK}) 00H: CKS020 bit and CKS021 bit specified operation clock (F_{MCK}) 01H: The active edge of the TI02 pin input signal Channel 0: The active edge of the input signal selected by TIOS0 Channel 1: The active edge of the input signal selected by TIOS0 Counting clocks (F_{TCLK}) are used in counters, output control circuits, and interrupt control circuits. | 0 |
| 11 | MASTER | Selection of independent channel operation/multi-channel operation (slave or master) for channel 2 00H: Used as a slave channel for independent | 0 |

| | | | |
|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | <p>or multi-channel operation.</p> <p>01H: Used as a master control channel for multi-channel operation.</p> <p>Channel 0 is fixed to "0" (since channel 0 is the highest bit channel, it is used as the master channel regardless of the setting of this bit).</p> <p>Only channel 2 can be set as the master channel (MASTERmn=1).</p> | |
| 10:8 | STS022- STS020 | <p>Start trigger and capture trigger settings for channel n</p> <p>STS022- STS020:</p> <p>00H: Only software triggering is active at the start (no other trigger source is selected).</p> <p>01H: Use the active edge of the TI02 pin input for start triggering and capture triggering.</p> <p>02H: Use the double edges of the TI02 pin input for start triggering and capture triggering respectively.</p> <p>04H: Use interrupt signals from the master channel (in the case of slave channels with multi-channel linkage operation function).</p> <p>Other than the above, settings are prohibited.</p> | 0x0 |
| 7:6 | CIS021- CIS020 | <p>Active edge selection for TI01 pins</p> <p>CIS021- CIS020</p> <p>00H: Falling edge</p> <p>01H: Rising edge</p> <p>02H: Double edge (when measuring low level width)</p> <p>Start trigger: falling edge, capture trigger: rising edge</p> <p>03H: Double edge (when measuring high level width)</p> <p>Start trigger: rising edge, capture trigger: falling edge</p> | 0x0 |
| 5:4 | 0 | Reserved | 0x0 |
| 3:0 | MD023- MD020 | <p>Setting of channel n operation mode and interrupt</p> <p>MD023-MD020:</p> <p>00H: Interval timer mode, no timer interrupt is generated at the start of counting.</p> <p>01H: Interval timer mode, a timer interrupt is generated when counting starts.</p> <p>04H: Capture mode, no timer interrupt is generated when counting starts.</p> <p>05H: Capture mode, no timer interrupt is generated when counting starts.</p> <p>06H: Event counter mode, no timer interrupt is generated when counting starts.</p> <p>08H: Single count mode, the start trigger in the count operation is invalid. No interruption at this time.</p> <p>09H: Single count mode, the start trigger in the</p> | |

| | | | |
|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------|--|
| | | count operation is valid. No interruption at this time. 0CH: Capture & single count mode, no timer interrupt is generated when counting starts. | |
|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------|--|

For a detailed description of MD023- MD020, see the following table

| MD 023 | MD 022 | MD 021 | Setting of channel n operation mode | Corresponding functions | Count operation of TCR |
|---------------------------------------------------------------------------------|--------|--------|-------------------------------------|--------------------------------------------------------------------------------------|------------------------|
| 0 | 0 | 0 | Interval timer mode | Interval timer/square wave output/ Frequency divider function/PWM output (master) | Count down |
| 0 | 1 | 0 | Capture mode | Measurement of input pulse interval | Count up |
| 0 | 1 | 1 | Event counter mode | External event counter | Count down |
| 1 | 0 | 0 | Single count mode | Delay counter/single trigger pulse output/PWM output (slave) | Count down |
| 1 | 1 | 0 | Capture & Single count mode | Measurement of the high- and low-level width of the input signal | Count up |
| Other than the above | | | Settings are prohibited. | | |
| The operation of each mode varies depending on MD020 bit (see the table below). | | | | | |

| Operation mode (Value set by the MD023 to MD021 bits (see table above)) | MD 020 | Setting of starting counting and interrupt |
|-----------------------------------------------------------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Interval timer mode (0, 0, 0) Capture mode (0, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| | 1 | A timer interrupt is generated when counting starts (the output of the timer also changes). |
| <ul style="list-style-type: none"> Event counter mode (0, 1, 1) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| <ul style="list-style-type: none"> Single count mode ^{Note 1} (1, 0, 0) | 0 | The start trigger in the count operation is invalid. No interruption at this time. |
| | 1 | The start trigger in the count operation is valid ^{Note 2} . No interruption at this time. |
| <ul style="list-style-type: none"> Capture & single count mode (1, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). The start trigger in the count operation is invalid. No interruption at this time. |

Note 1: In single count mode, the interrupt output (INTTM02) and TO02 output at the start of counting are not controlled.

Note 2: If a start trigger is generated during operation (TS02=1), the counter is initialized and counting is restarted (no interrupt request is generated).

Table 6-14: Timer channel 3 mode register TMR03

| Bit | Symbol | Description | Reset value |
|-------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:14 | CKS031- CKS030 | Selection of channel n operation clock (F_{MCK}) CKS031- CKS030: 00H: The operation clock CKm0 set by the timer clock select register m (TPSm) 01H: The operation clock CKm2 set by the timer clock select register m (TPSm) 02H: The operation clock CKm1 set by the timer clock select register m (TPSm) 03H: The operation clock CKm3 set by the timer clock select register m (TPSm) The operation clock (F_{MCK}) is used for edge detection circuits. The sample clock and count clock (F_{TCLK}) are generated by setting the CCSmn bit. Only Channel 1 and Channel 3 can select operation clocks CKm2 and CKm3. | 0x0 |
| 13 | 0 | Must be set to 0 | 0 |
| 12 | CCS03 | Selection of channel n count clock (F_{TCLK}) 00H: CKS030 bit and CKS031 bit specified operation clock (F_{MCK}) 01H: The active edge of the TI03 pin input signal Channel 0: The active edge of the input signal selected by TIOS0 Channel 1: The active edge of the input signal selected by TIOS0 Counting clocks (F_{TCLK}) are used in counters, output control circuits, and interrupt control circuits. | 0 |
| 11 | SPLIT0 | Operation selection of 8-bit timer/16-bit timer for channel 3 00H: Used as a 16-bit timer. (Used as a slave channel for independent channel operation or multi-channel linkage operation) 01H: Used as an 8-bit timer. | 0 |
| 10:8 | STS032- STS030 | Start trigger and capture trigger settings for channel n STS032- STS030: 00H: Only software triggering is active at the start (no other trigger source is selected). 01H: Use the active edge of the TI03 pin input for start triggering and capture triggering. 02H: Use the double edges of the TI03 pin input for start triggering and capture triggering respectively. 04H: Use interrupt signals from the master channel (in the case of slave channels with multi-channel linkage operation function). Other than the above, settings are prohibited. | 0x0 |
| 7:6 | CIS031- CIS030 | Active edge selection for TI01 pins CIS031- CIS030 00H: Falling edge 01H: Rising edge 02H: Double edge (when measuring low level width) Start trigger: falling edge, capture trigger: | 0x0 |

| | | | |
|-----|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | rising edge 03H: Double edge (when measuring high level width) Start trigger: rising edge, capture trigger: falling edge | |
| 5:4 | 0 | Reserved | 0x0 |
| 3:0 | MD033- MD030 | Setting of channel n operation mode and interrupt MD033-MD030: 00H: Interval timer mode, no timer interrupt is generated at the start of counting. 01H: Interval timer mode, a timer interrupt is generated when counting starts. 04H: Capture mode, no timer interrupt is generated when counting starts. 05H: Capture mode, no timer interrupt is generated when counting starts. 06H: Event counter mode, no timer interrupt is generated when counting starts. 08H: Single count mode, the start trigger in the count operation is invalid. No interruption at this time. 09H: Single count mode, the start trigger in the count operation is valid. No interruption at this time. 0CH: Capture & single count mode, no timer interrupt is generated when counting starts. | |

For a detailed description of MD033- MD030, see the following table

| MD 033 | MD 322 | MD 031 | Setting of channel n operation mode | Corresponding functions | Count operation of TCR |
|---------------------------------------------------------------------------------|--------|--------|-------------------------------------|--------------------------------------------------------------------------------------|------------------------|
| 0 | 0 | 0 | Interval timer mode | Interval timer/square wave output/ Frequency divider function/PWM output (master) | Count down |
| 0 | 1 | 0 | Capture mode | Measurement of input pulse interval | Count up |
| 0 | 1 | 1 | Event counter mode | External event counter | Count down |
| 1 | 0 | 0 | Single count mode | Delay counter/single trigger pulse output/PWM output (slave) | Count down |
| 1 | 1 | 0 | Capture & Single count mode | Measurement of the high- and low-level width of the input signal | Count up |
| Other than the above | | | Settings are prohibited. | | |
| The operation of each mode varies depending on MD030 bit (see the table below). | | | | | |

| Operation mode (Value set by the MD023 to MD021 bits (see table above)) | MD030000 | Setting of starting counting and interrupt |
|-----------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Interval timer mode (0, 0, 0) Capture mode (0, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). |
| | 1 | A timer interrupt is generated when counting starts (the output of the timer also changes). |
| <ul style="list-style-type: none"> Event counter mode (0, 1, 1) | 0 | No timer interrupt is generated when counting starts (the |

| | | |
|-------------------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | output of the timer does not change). |
| <ul style="list-style-type: none"> Single count mode ^{Note 1} (1, 0, 0) | 0 | The start trigger in the count operation is invalid. No interruption at this time. |
| | 1 | The start trigger in the count operation is valid ^{Note 2} . No interruption at this time. |
| <ul style="list-style-type: none"> Capture & single count mode (1, 1, 0) | 0 | No timer interrupt is generated when counting starts (the output of the timer does not change). The start trigger in the count operation is invalid. No interruption at this time. |

Note 1: In single count mode, the interrupt output (INTTM03) and TO03 output at the start of counting are not controlled.

Note 2: If a start trigger is generated during operation (TS03=1), the counter is initialized and counting is restarted (no interrupt request is generated).

6.2.7 Timer status register mn (TSRmn)

The TSRmn register is a register that indicates the overflow status of the channel n counter.

The TSRmn register is valid only in capture mode (MDmn3~MDmn1=010B) and capture & single count mode (MDmn3~MDmn1=110B). Refer to Table 6-16 for the OVF bit changes and set/clear conditions in each operation mode.

The TSRmn register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be read with TSRmnL and 8-bit memory manipulation instructions. After a reset signal is generated, the value of the TSRmn register changes to “0000H”.

Table 6-15: Table of timer status register mn (TSRmn)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:1 | - | Reserved | - |
| 0 | OVF | Counter overflow status of channel n 0: No overflow occurred. 1: Overflow occurred. If the OVF bit is “1”, this flag is cleared when the next count does not overflow and the count value is captured (OVF=0). | 0 |

Note: m: unit number (m=0) n: channel number (n=0~3)

Table 6-16: OVF bit change and set/clear conditions in each operation mode

| Timer operation mode | OVF bit | Set/clear conditions |
|-------------------------------|---------|--------------------------------------|
| • Capture mode | Clear | No overflow occurred at the capture. |
| • Capture & single count mode | Set | Overflow occurred at the capture. |
| • Interval timer mode | Clear | — (N/A) |
| • Event counter mode | Set | |
| • Single count mode | | |

Note: Even if the counter overflows, the OVF bit does not change immediately, but changes on subsequent captures.

6.2.8 Timer channel enable status register m (TEm)

The TEm register is a register that indicates the enable or stop status of each channel timer operation.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and timer channel stop register m (TTm). If each bit of the TSm register is “1”, the corresponding bit of the TEm register is “1”. If each bit of the TTm register is “1”, the corresponding bit of the TEm register is cleared to “0”.

The TEm register is read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be read with TEmL and 8-bit memory manipulation instructions. After a reset signal is generated, the value of the TEm register changes to “0000H”.

Table 6-17: Table of timer channel enable status register m (TEm)

| Bit | Symbol | Description | Reset value |
|-------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:12 | - | Reserved | - |
| 11 | TEH03 | Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode 0: Operation is stopped 1: Operation is enabled | - |
| 10 | - | Reserved | 0 |
| 9 | TEH01 | Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode 0: Operation is stopped 1: Operation is enabled | 0 |
| 8:4 | - | Reserved | 0x0 |
| 3:0 | TE03-TE00 | Indication of operation enable/stop status of channel n 0: Operation is stopped 1: Operation is enabled | 0x0 |

6.2.9 Timer channel start register m (TSm)

The TSm register is a trigger register that initializes the timer counter register mn (TCRmn) and sets the start of counting operation for each channel. If each bit is set to “1”, the corresponding bit of the timer channel enable status register m (TEm) is set to “1”. Since the TSmn bit, the TSHm1 bit and the TSHm3 bit are trigger bits, the TSmn bit, the TSHm1 bit and the TSHm3 bit are cleared immediately if the operation enable state is changed (TEmn, TEHm1, TEHm3 = 1).

The TSm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set by TSmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TSm register changes to “0000H”.

Table 6-18: Table of timer channel start register m (TSm)

| Bit | Symbol | Description | Reset value |
|-------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:12 | - | Reserved | - |
| 11 | TSHm3 | Trigger to enable (start) operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode 0: No trigger operation. 1: Set the TEHm3 bit to “1” to enter the counting enable state. If the counting of the TCRm3 register is started in the count enable state, the interval timer mode is entered (refer to Table 6-26 of “6.4.2 Start Timing of Counter”). | - |
| 10 | - | Reserved | 0 |
| 9 | TSHm1 | Trigger to enable (start) operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode 0: No trigger operation 1: Set the TEHm1 bit to “1” to enter the counting enable state. If counting in the TCRm1 register is started in the count enable state, the interval timer mode is entered (refer to Table 6-26 of “6.4.2 Start Timing of Counter”). | 0 |
| 8:4 | - | Reserved | 0x0 |
| 3:0 | TSm3-TSm0 | Operation enable (start) trigger of channel n 0: No trigger operation 1: Set the TEmn bit to “1” to enter the counting enable state. The start of counting in the TCRmn register in the count enable state varies with each operation mode (refer to Table 6-26 of “6.4.2 Start Timing of Counter”). When channel 1 and channel 3 are in 8-bit timer mode, TSm1 and TSm3 are operation enable (start) triggers for the lower 8-bit timer. | 0x0 |

Note 1: Bits 15~12, 10, 8~4 must be set to “0”.

Note 2: When switching from a function that does not use TImn pin input to a function that uses TImn pin input, the following period of waiting is required from setting the timer mode register mn (TMRmn)

until the TSmn bit is set to “1”:

When the TImn pin noise filter is valid (TNFENmn=1): 4 operating clocks (F_{MCK})

When the TImn pin noise filter is invalid (TNFENmn=0): 2 operating clocks (F_{MCK})

Note 3: The TSm register always reads “0”.

Note 4: m: unit number (m=0)

6.2.10 Timer channel stop register m (TTm)

The TTm register is a trigger register to set the count stop of each channel.

If each bit is set to “1”, the corresponding bit in the timer channel enable status register m (TEm) is cleared to “0”. Since the TTmn bit, TTHm1 bit, and TTHm3 bit are trigger bits, the TTmn bit, TTHm1 bit, and TTHm3 bit are cleared immediately if the operation stop state is changed (TEmn, TEHm1, and TEHm3 = 0).

The TTm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set by TTmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TTm register changes to “0000H”.

Table 6-19: Table of timer channel stop register m (TTm)

| Bit | Symbol | Description | Reset value |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:12 | - | Reserved | - |
| 11 | TTHm3 | Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode 0: No trigger operation 1: TEHm3 bit is cleared to 0 and the count operation is stopped. | - |
| 10 | - | Reserved | 0 |
| 9 | TTHm1 | Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode 0: No trigger operation 1: TEHm1 bit is cleared to 0 and the count operation is stopped. | 0 |
| 8:4 | - | Reserved | 0x0 |
| 3:0 | TTm3-TTm0 | Operation stop trigger of channel n 0: No trigger operation 1: TEMn bit clear to 0, to be count operation stop enable status. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode. | 0x0 |

Note 1: Bits 15~12, 10, 8~4 must be set to “0”.

Note 2: The TTm register always reads “0”.

Note 3: m: unit number (m=0)

6.2.11 Timer input/output output select register (TIOS0)

The TIOS0 register is used to make selections for the inputs and outputs of unit 0. The timer inputs for channel 0 and channel 1 and the timer output for channel 2 of unit 0 are selected. The TIOS0 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TIOS0 register changes to “00H”.

Table 6-20: Table of timer input/output select register 0 (TIOS0)

| Bit | Symbol | Description | Reset value |
|-----|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:5 | TIS07- TIS05 | Selection of timer input used for channel 0 0: Input signal for timer input pin (TI00) Other: Settings are prohibited. | - |
| 4 | TIS04 | Selection of timer input used for channel 0 0: Input signal selected by TIS07~TIS05 | 0 |
| 3 | TOS03 | Enable channel 2 timer output 0: Output enable 1: Output disable (output fixed to 0) | 0 |
| 2:0 | TIS02- TIS00 | Selection of timer input used for channel 1 00H: Input signal for timer input pin (TI01) 02H: Input signal for timer input pin (TI01) 03H: Input signal for timer input pin (TI01) 04H: Low-speed on-chip oscillator clock (F _{IL}) Other than the above, settings are prohibited. | 0x0 |

Note 1: The high-/low-level width of the selected timer inputs needs to be greater than or equal to $1/F_{MCK}+10\text{ns}$. Therefore, when F_{IL} is selected as the F_{CLK} (CSS bit of the CKC register =1), the TIS02 bit cannot be set to “1”.

6.2.12 Timer output enable register m (TOEm)

The TOEm register is a register that sets to enable or disable the timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOMn bit of timer output register m (TOM) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOMn).

The TOEm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set by TOEmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOEm register changes to “0000H”.

Table 6-21: Table of timer output enable register m (TOEm)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:4 | | Must be set to 0 | 0x0 |
| 3:0 | TOEmn | Enable/disable the timer output of channel n 0: Disable timer output. The operation of the timer is not reflected to the TOMn bit, fixed output. The TOMn bit can be written and the level set by the TOMn bit is output from the TOMn pin. 1: Enable timer output The operation of the timer is reflected to the TOMn bit, producing an output waveform. The write of the TOMn bit is ignored. | 0x0 |

Note 1: Bits 15~4 must be set to “0”.

Note 2: m: unit number (m=0) n: channel number (n=0~3)

6.2.13 Timer output register m (TOM)

The TOM register is a buffer register for each channel timer output.

The bit value of this register is output from the output pin (TOMn) of each channel timer.

The TOMn bit of this register can be rewritten by software only when timer output is disabled (TOEmn=0). When enabling the timer output (TOEmn=1), rewrite operations via software are ignored and its value is changed only by the operation of the timer.

To use the TI00/TO00, TI01/TO01, TI02/TO02, and TI03/TO03 pins as port functions, the corresponding TOMn bit must be set to "0".

The TOM register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM register can be set by TOML and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOM register changes to "0000H".

Table 6-22: Table of timer output register m (TOM)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------------------------------|-------------|
| 15:4 | | Must be set to 0 | 0x0 |
| 3:0 | TOMn | Timer output of channel n 0: The output value of the timer is "0". 1: The output value of the timer is "1". | 0x0 |

Note: m: unit number (m=0) n: channel number (n=0~3)

6.2.14 Timer output level register m (TOLm)

The TOLm register is a register that controls the output level of each channel timer.

When timer output (TOEmn=1) is enabled and the multi-channel linkage operation function (TOMmn=1) is used, the set and reset timing of the timer output signal reflects the inverse setting of each channel n performed by this register. In the master channel output mode (TOMmn=0), this register setting is invalid.

The TOLm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set by TOLmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOLm register changes to "0000H".

Table 6-23: Table of timer output level register m (TOLm)

| Bit | Symbol | Description | Reset value |
|------|--------------|-------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:4 | | Must be set to 0 | 0x0 |
| 3:1 | TOL03- TOL01 | Control of timer output level of channel n 0: Positive logic output (active-high) 1: Inverted output (active-low) | 0x0 |
| 0 | 0 | Reserved to 0 | 0 |

Note 1: If the value of this register is rewritten while the timer is operating, the timer output logic is inverted at the next time the timer output signal changes, rather than immediately after the rewrite.

Note 2: m: unit number (m=0) n: channel number (n=0~3)

6.2.15 Timer output mode register m (TOMm)

The TOMm register is a register that controls the output mode of each channel timer. When used as an independent channel operation function, the corresponding bit of the using channel should be set to “0”.

When used as a multi-channel linkage operation function (PWM output, single trigger pulse output and multiple PWM output), the corresponding bit of the master channel is “0” and the corresponding bit of the slave channel is “1”.

When the timer output (TOEmn=1) is enabled, the setting of each channel n is reflected in this register during the setting and resetting timing of the timer output signal.

The TOMm register is set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set by TOMmL and by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the TOMm register changes to “0000H”.

Table 6-24: Table of timer output mode register m (TOMm)

| Bit | Symbol | Description | Reset value |
|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 15:4 | | Must be set to 0 | 0x0 |
| 3:1 | TOM03- TOM01 | Control of channel n timer output mode 0: Master channel output mode (toggle output via timer interrupt request signal (INTTMmn)) 1: Slave channel output mode (output is set via timer interrupt request signal (INTTMmn) of master channel and output is reset via timer interrupt request signal (INTTMmp) of slave channel) | 0x0 |
| 0 | 0 | Reserved to 0 | 0 |

Note: m: unit number (m=0) n: channel number n=0~3 (master channel: n=0, 2)

p: slave channel

n=0: p=1, 2, 3

n=2: p=3

(For details on the relationship between the master channel and the slave channel, refer to “6.3.1 Basic Rules for Multi-Channel Linkage Operation Function”).

6.2.16 Noise filter enable register 1 (NFEN1)

The NFEN1 register sets whether the noise filter is used for the input signals of the timer input pins of each channel of Unit 0. For pins that require noise removal, the corresponding bit must be set to “1” to make the noise filter effective. When the noise filter is enabled, after synchronization with the operating clock (F_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (F_{MCK}) for the target channel^{Note}.

The NFEN1 register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of the NFEN1 register changes to “00H”.

Note: For details, refer to “6.4.1(2) When the valid edge of TImn pin input signal (CCSmn=1) is selected”, “6.4.2 Start Timing of Counter”, and “6.6 Control of Timer Input (TImn)”.

Table 6-25: Table of noise filter enable register 1 (NFEN1)

| Bit | Symbol | Description | Reset value |
|-----|---------|---------------------------------------------------------------------------------------------|-------------|
| 7:4 | - | Reserved | 0x0 |
| 3 | TNFEN03 | Usage of input signal noise filter on TI03 pin 0: Noise filter OFF 1: Noise filter ON | 0 |
| 2 | TNFEN02 | Usage of input signal noise filter on TI02 pin 0: Noise filter OFF 1: Noise filter ON | 0 |
| 1 | TNFEN01 | Usage of input signal noise filter on TI01 pin 0: Noise filter OFF 1: Noise filter ON | 0 |
| 0 | TNFEN00 | Usage of input signal noise filter on TI00 pin 0: Noise filter OFF 1: Noise filter ON | 0 |

Note: Refer to “Chapter 3 Pin Function” for the configuration of timer input/output pins of channels 0~3.

6.2.17 Registers controlling port functions of timer input/output pins

When using the General-Purpose Timer Unit, the output pins of Timer0 are multiplexed to a fixed port, and the input pins of Timer0 can be configured to any port. For details, refer to “Chapter 3 Pin Function”.

When multiplexing the output pin of Timer 0 to a port, the corresponding bit of the Port Mode Control Register (PMCxx), the bit of the Port Mode Register (PMxx), and the bit of the Port Register (Pxx) must be set to “0”.

When multiplexing the output pin of Timer 0 to a port, the bit of the Port Mode Control Register (PMCxx) corresponding to that port, and the bit of the Port Mode Register (PMxx) must be set to “0”. The port multiplexing function configuration register (PxxCFG) is also set. In this case, the bit of the port register (Pxx) can be “0” or “1”.

(Example)

When P20 is configured as TO00 and used as a timer output

Set the PMC20 bit of port mode control register 2 to “0”.

Set bit PM20 of port mode register 2 to “0”.

Set port output multiplexing function configuration register P20CFG to “0x0a”.

When using the multiplexed port of the Timer0 input pin as the timer input, the corresponding bit of the Port Mode Register (PMx) is set to “1”, the bit of the Port Mode Control Register (PMCxx) is set to “0” and set the Port Mode Configuration Register (PStau0tin0_CFG). In this case, the bit of the port register (Pxx) can be “0” or “1”.

(Example)

Using P20/TI00 as a timer input.

Set the bit PMC20 of the Port Mode Control Register 2 to “0”.

Set the bit PM20 of the Port Mode Register 2 to “1”.

Set port input multiplexing function configuration register PStau0tin0_CFG to 0x20.

6.3 Basic rules of general-purpose timer unit

6.3.1 Basic rules of multi-channel linkage operation function

The multi-channel linkage function is a function that combines a master channel (a reference timer that counts cycles) and a slave channel (a timer that operates in compliance with the master channel), and several rules need to be observed when using it.

The basic rules of the multi-channel linkage operation function are shown below.

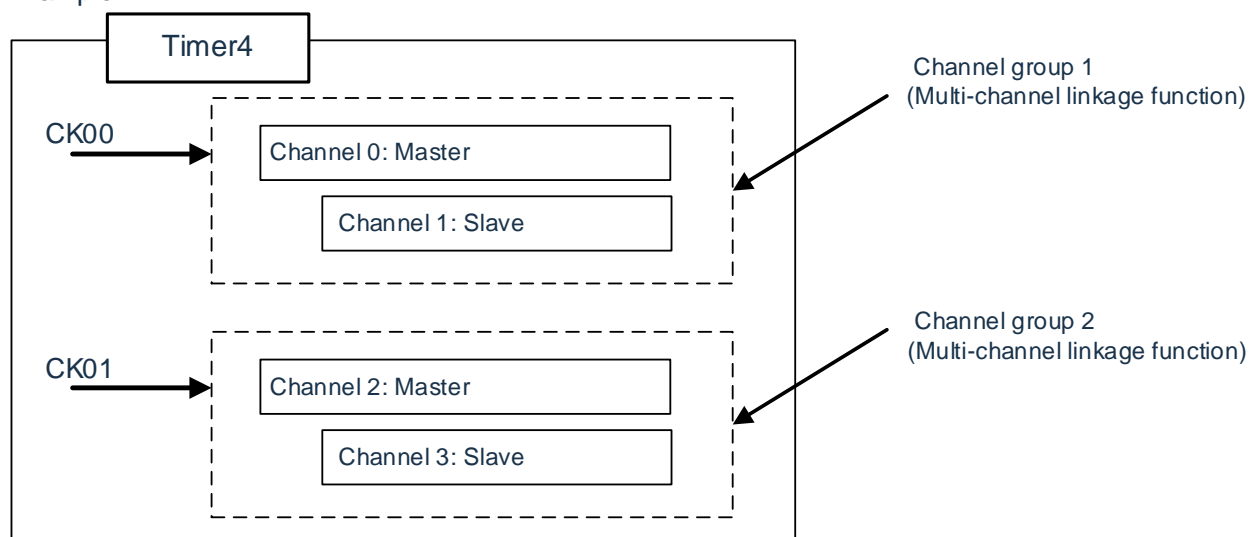
- (1) Only the even-number channel (channel 0, channel 2) can be set as a master channel.
- (2) Any channel other than channel 0 can be set as a slave channel.
- (3) Only the lower channel of the master channel can be set as a slave channel.
For example, when setting channel 0 as the master channel, it is possible to set the channels starting from channel 1 (channels 1 to 3) as slave channels.
- (4) Multiple slave channels can be set for 1 master channel.
- (5) When multiple master channels are used, slave channels that span the master channel cannot be set.
For example, when setting channel 0 and channel 2 as the master channel, channel 1 can be set as the slave channel of master channel 0, but channel 3 cannot be set as the slave channel of master channel 0.
- (6) The slave channels linked to the master channel need to be set to the same operating clock. The CKSmn0 bit and CKSmn1 bit (bit15 and bit14 of Timer Mode Register mn (TMRmn)) of the slave channel linked to the master channel need to be the same setting value.
- (7) The master channel can pass the INTTMmn (interrupt), start software trigger and count clock to the lower channel.
- (8) The slave channel can use the master channel's INTTMmn (interrupt), start software trigger, and count clocks as source clocks, but cannot pass its own INTTMmn (interrupt), start software trigger, and count clocks to the lower channel.
- (9) The master channel cannot use the INTTMmn (interrupt), start software trigger and count clocks of other high master channels as source clocks.
- (10) In order to start the channels to be linked at the same time, the channel start trigger bit (TSmn) of the linked channel needs to be set at the same time.
- (11) Only all linked channels or the master channel can use the setting of the TSmn bit in the counting operation. It is not possible to use the setting of the TSmn bit of the slave channel only.
- (12) In order to stop the linked channels at the same time, the channel stop trigger bit (TTmn) of the linked channel needs to be set at the same time.
- (13) In linked operation, CKm2/CKm3 cannot be selected because the master and slave channels need the same operating clock.
- (14) The timer mode register m0 (TMRm0) has no master bit and is fixed to "0". However, since channel 0 is the highest bit channel, it can be used as the master channel during linkage operation.

The basic rules of the multi-channel linkage operation function are the rules applicable to the group of channels (a collection of master and slave channels that form a multi-channel linkage operation function).

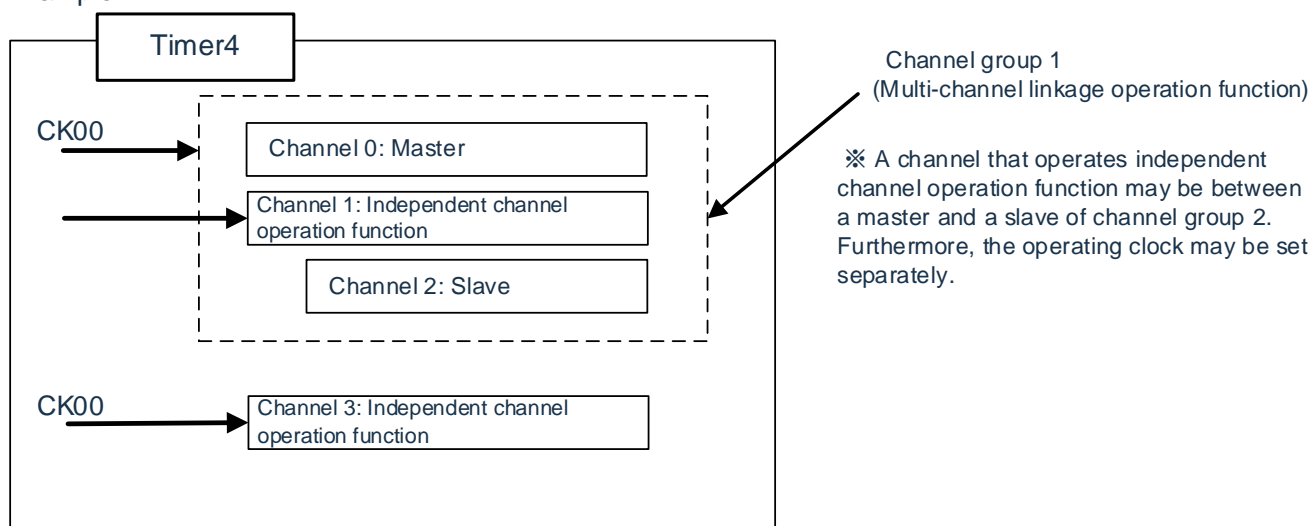
If you set 2 or more channel groups that are not linked to each other, the above basic rules do not apply to the channel groups.

Note: m: unit number (m=0) n: channel number (n=0~3)

Example 1



Example 2



6.3.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

- (1) The 8-bit timer operation function applies only to channels 1 and 3.
- (2) When using 8-bit timers, set the SPLIT bit of timer mode register mn (TMRmn) to 1.
- (3) The higher 8 bits can be operated as the interval timer function.
- (4) At the start of operation, the higher 8 bits output INTTMm1H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
- (5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
- (6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
- (7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
 - Interval timer function
 - External event counter function
 - Delay count function
- (8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.
- (9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.
- (10) For the 8-bit timer function, the linkage operation functions (single pulse, PWM, and multiple PWM) cannot be used.

Note: unit number (m=0) n: channel number (n=1, 3)

6.4 Operation of counter

6.4.1 Count clock (F_{TCLK})

The count clock of the general-purpose timer unit (F_{TCLK}) can be selected by the CCSmn bit of the timer mode register mn (TMRmn) for any of the following clocks:

- ① The CKSmn0 bit and CKSmn1 bit specified operation clock (F_{MCK})
- ② The active edge of the TImn pin input signal

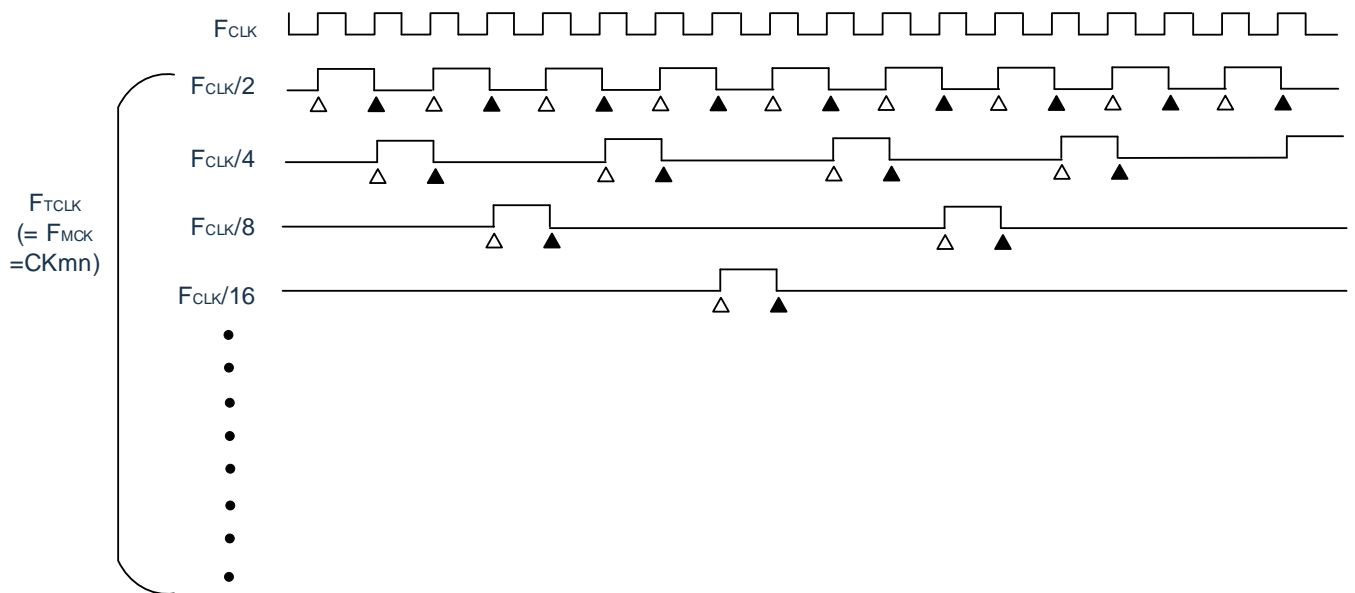
The general-purpose timer unit is designed to operate synchronously with F_{CLK} , so the timing of the count clock (F_{TCLK}) is as follows.

- (1) When operation clock (F_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

According to the setting of timer clock selection register m (TPSm), the counting clock (F_{TCLK}) is $F_{CLK} \sim F_{CLK} / 2^{15}$. However, when the frequency division of F_{CLK} is selected, the clock selected by TPSm register is a signal that has only 1 F_{CLK} cycle of high level from the rising edge. When F_{CLK} is selected, it is fixed to high level.

In order to obtain synchronization with F_{CLK} , timer count register mn (TCRmn) delays the counting by one F_{CLK} clock from the rising edge of the counting clock, which is called “counting at the rising edge of the counting clock” for convenience.

Figure 6-2: Timing of F_{CLK} and count clock (F_{TCLK}) (When CCSmn = 0)



Note 1: Δ : Rising edge of the count clock

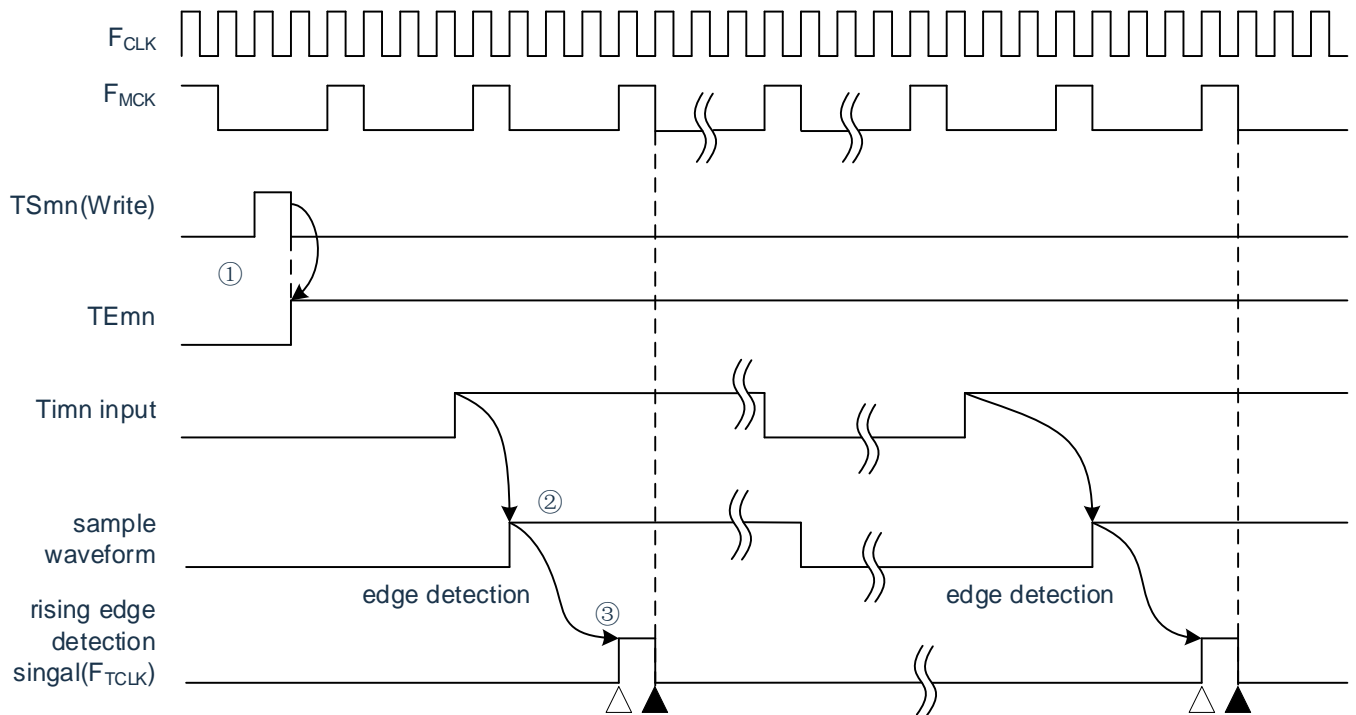
Note 2: \blacktriangle : Synchronization, increment/decrement of counter

Note 3: F_{CLK} : CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (F_{TCLK}) is a signal that detects an active edge of the TImn pin input signal and is synchronized with the next F_{MCK} rising edge. In fact, this is a signal delayed by 1~2 F_{MCK} clocks compared to the input signal of the TImn pin (delay 3~4 F_{MCK} clocks when using noise filters). In order to obtain synchronization with F_{CLK} , the timer count register mn (TCRmn) delays the count by one F_{CLK} time from the rising edge of the count clock, which is referred to as "counting at the effective edge of the TImn pin input signal" for convenience.

Figure 6-3: Timing of the counting clock (F_{TCLK}) (CCSmn=1, without noise filter)



- ① Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.
- ② The rise of input signal via the TImn pin is sampled by F_{MCK} .
- ③ The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Note 1: \triangle : Rising edge of the count clock

Note 2: \blacktriangle : Synchronization, increment/decrement of counter

Note 3: F_{CLK} : CPU/peripheral hardware clock

Note 4: F_{MCK} : Operation clock of channel n

Note 5: The same waveforms are used for the measurement of the input pulse interval, the high and low measurement of the input signal, the delay counter and the TImn input for the single trigger pulse output function.

6.4.2 Start timing of counter

The timer count register mn (TCRmn) enters the operation enable state by setting TSmn bit of the timer channel start register m (TSM).

Execution from the counting enable state to the start of the timer count register mn (TCRmn) is shown in Table 6-26.

Table 6-26: Operation from the counting enable state to the start of the timer count register mn (TCRmn)

| Timer operation mode | Operation after setting TSmn bit to "1" |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| • Interval timer mode | No operation is performed from the detection of the start trigger (TSmn=1) until the count clock is generated. The value of the TDRmn register is loaded into the TCRmn register by the first count clock and decremented by subsequent count clocks (refer to "6.4.3 Operation of the interval timer mode"). |
| • Event counter mode | The value of the TDRmn register is loaded into the TCRmn register by writing a "1" to the TSmn bit. If the input edge of TImn is detected, the count is decremented by the subsequent count clocks. (Refer to "6.4.3 Operation of the event counter mode"). |
| • Capture mode | No operation is performed from the time the start trigger is detected until the count clock is generated. The "0000H" is loaded into the TCRmn register by the first count clock, and incremental counting is performed by the subsequent count clocks (refer to "6.4.3 Operation of the capture mode (input pulse interval measurement)"). |
| • Single count mode | By writing "1" to the TSmn bit while the timer is stopped (TEmn=0), it enters the wait state for the start of the trigger. No operation is performed from the time the start trigger is detected until the count clock is generated. The value of the TDRmn register is loaded into the TCRmn register by the first count clock, and decremental counting by subsequent count clocks (refer to "6.4.3 Operation of the single count mode"). |
| • Capture & single count mode | By writing "1" to the TSmn bit while the timer is stopped (TEmn=0), it enters the wait state for the start of the trigger. No operation is performed from the time the start trigger is detected until the count clock is generated. The "0000H" is loaded into the TCRmn register by the first count clock, and incremental counting is performed by the subsequent count clocks (refer to "6.4.3 Operation of capture & single count mode (measurement of high-level width)"). |

6.4.3 Operation of counter

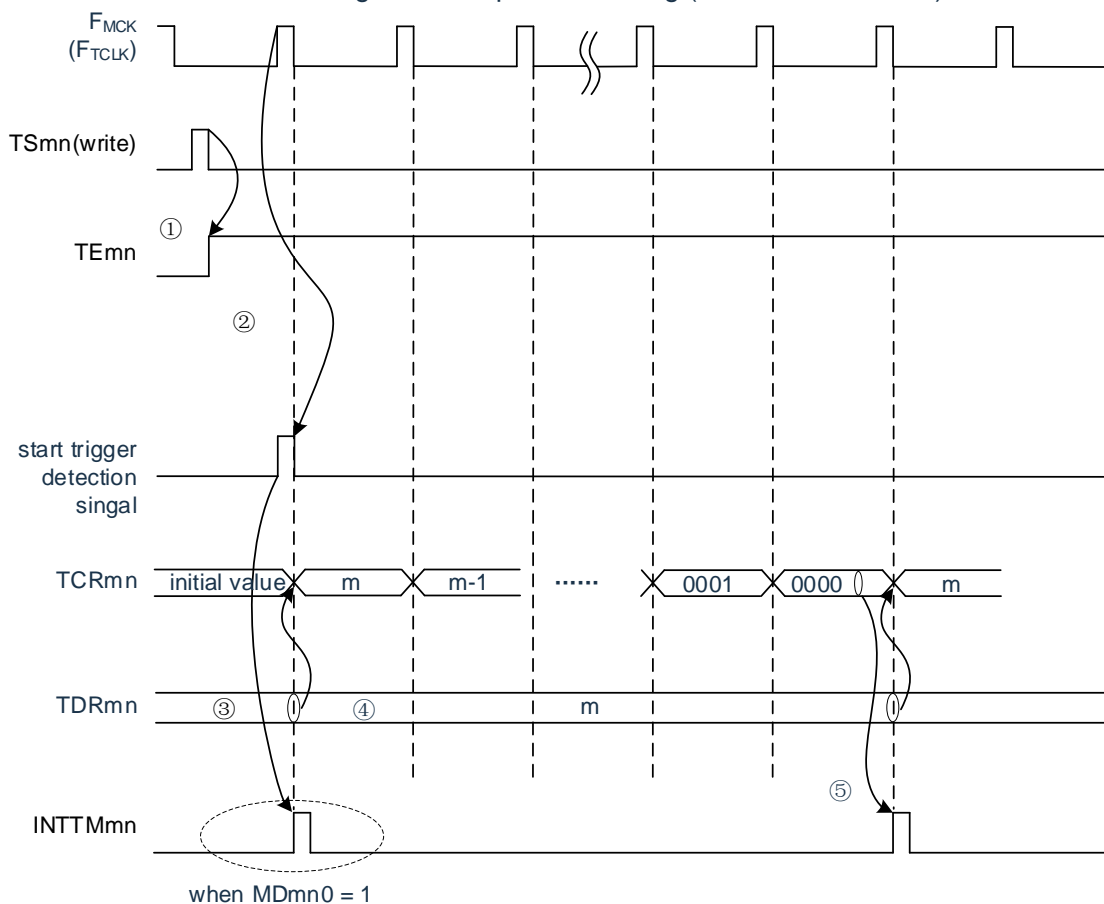
The following describes the counter operation for each mode.

(1) Operation of interval timer mode

- ① The operation enable state is entered by writing “1” to the TSmn bit (TEmn=1). The timer count register mn (TCRmn) remains at its initial value until a count clock is generated.
- ② A start trigger signal is generated by enabling the 1st count clock (F_{MCK}) after the operation.
- ③ When MDmn0 bit is “1”, INTTMmn is generated by the start trigger signal.
- ④ The value of timer data register mn (TDRmn) is loaded into the TCRmn register by enabling the 1st count clock after the operation, and counting starts in interval timer mode.

If the TCRmn register decrements to “0000H”, INTTMmn is generated by the next count clock (F_{MCK}) and continues counting after loading the value of timer data register mn (TDRmn) into the TCRmn register.

Figure 6-4: Operation timing (interval timer mode)



Note 1: In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

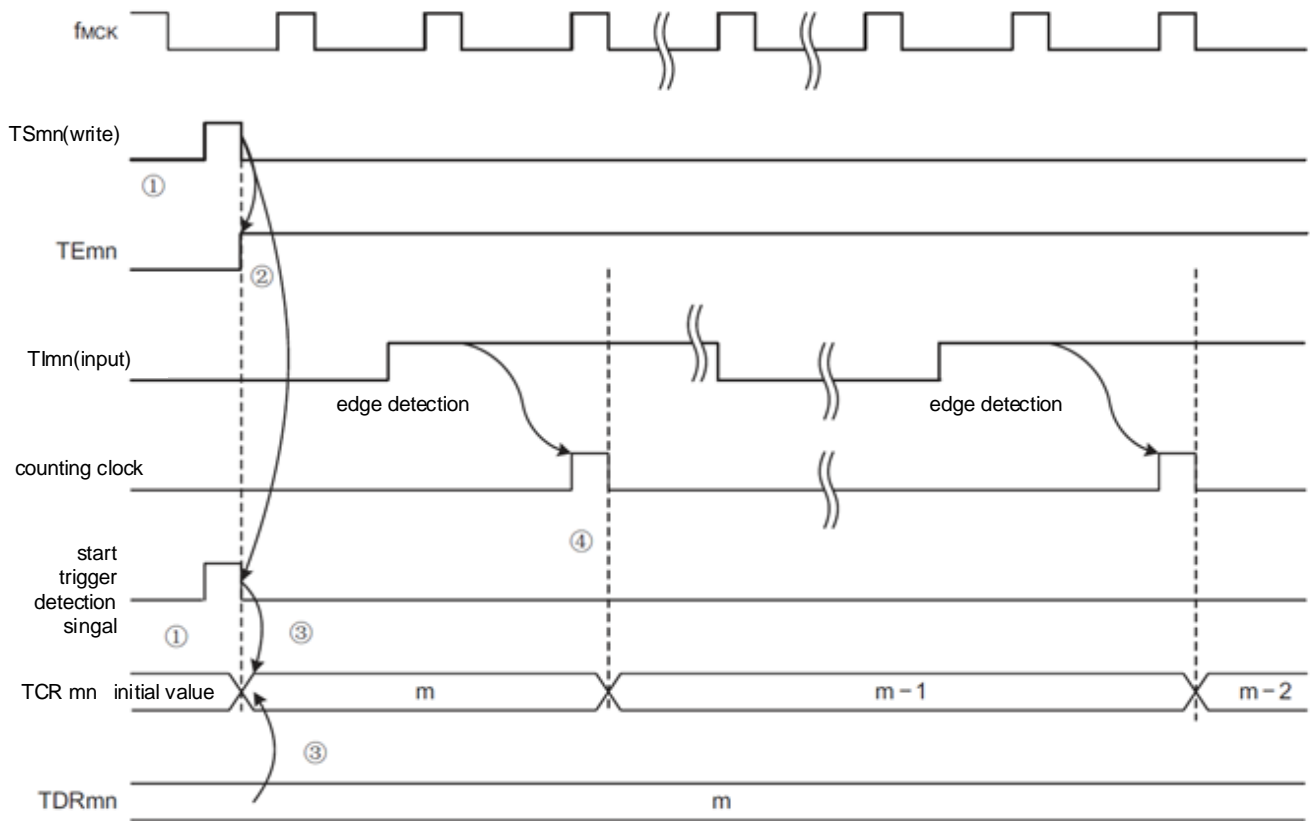
Note 2: F_{MCK} , the start trigger detection signal, and INTTMmn become active between one clock in synchronization with F_{CLK} .

(2) Operation of event counter mode

- ① The timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn=0).
- ② The operation enable state is enabled by writing "1" to the TSmn bit (TEmn=1).
- ③ The value of timer data register mn (TDRmn) is loaded into the TCRmn register while both the TSmn and TEMn bits are changed to "1" and counting begins.

Thereafter, the value of the TCRmn register is counted decreasingly by the count clock at the active edge of the TImn input.

Figure 6-5: Operation timing (event counter mode)

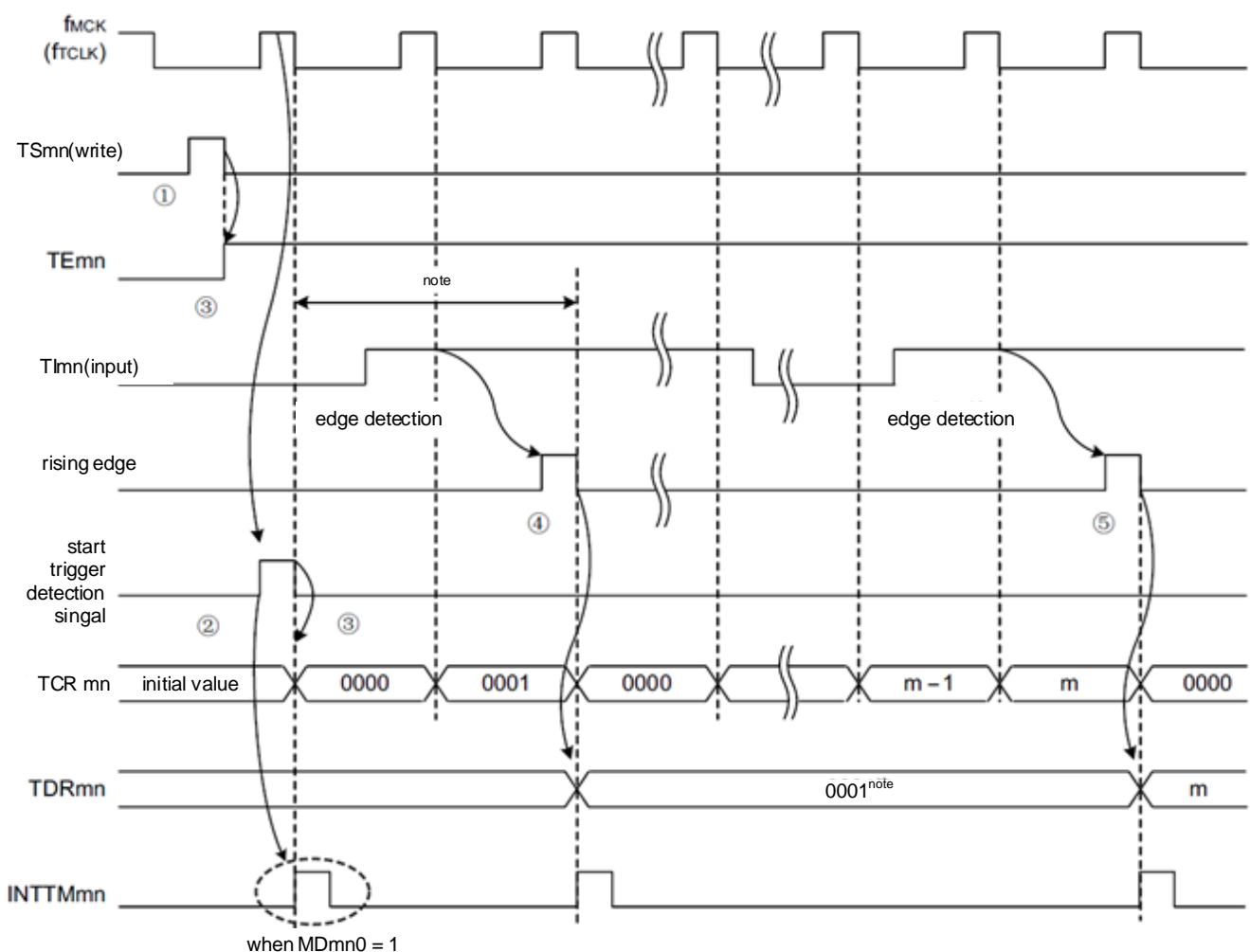


Note: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TImn input. The 1 cycle error is because the TImn input is not synchronized with the count clock (F_{MCK}).

(3) Operation of capture mode (interval measurement of input pulses)

- ① The operation enable state is entered by writing “1” to the TSmn bit (TEmn=1).
- ② The timer count register mn (TCRmn) remains at its initial value until a count clock is generated.
- ③ A start trigger signal is generated by enabling the 1st count clock (F_{MCK}) after the operation. Then, the “0000H” is loaded into the TCRmn register and counting starts in capture mode (INTTMmn is generated by the start trigger signal when MDmn0 bit is “1”).
- ④ If an active edge of TImn input is detected, the value of TCRmn register is captured to TDRmn register and INTTMmn interrupt is generated. The capture value is meaningless at this point. The TCRmn register continues counting from the “0000H”.
- ⑤ If an active edge of the next TImn input is detected, the value of the TCRmn register is captured to the TDRmn register and the INTTMmn interrupt is generated.

Figure 6-6: Operation timing (capture mode: interval measurement of input pulses)



Note: When the clock is input to TImn (with trigger) before the start, the count is started by detecting the trigger even if no edge is detected, so the capture value at the 1st capture (④) is not a pulse interval (in this example, 0001: 2 clock intervals) and must be ignored.

Note 1: The 1st count clock cycle runs after the TSmn bit is written and delays the start of counting before generating the count clock, an error of up to 1 clock cycle is generated. Also, if you need

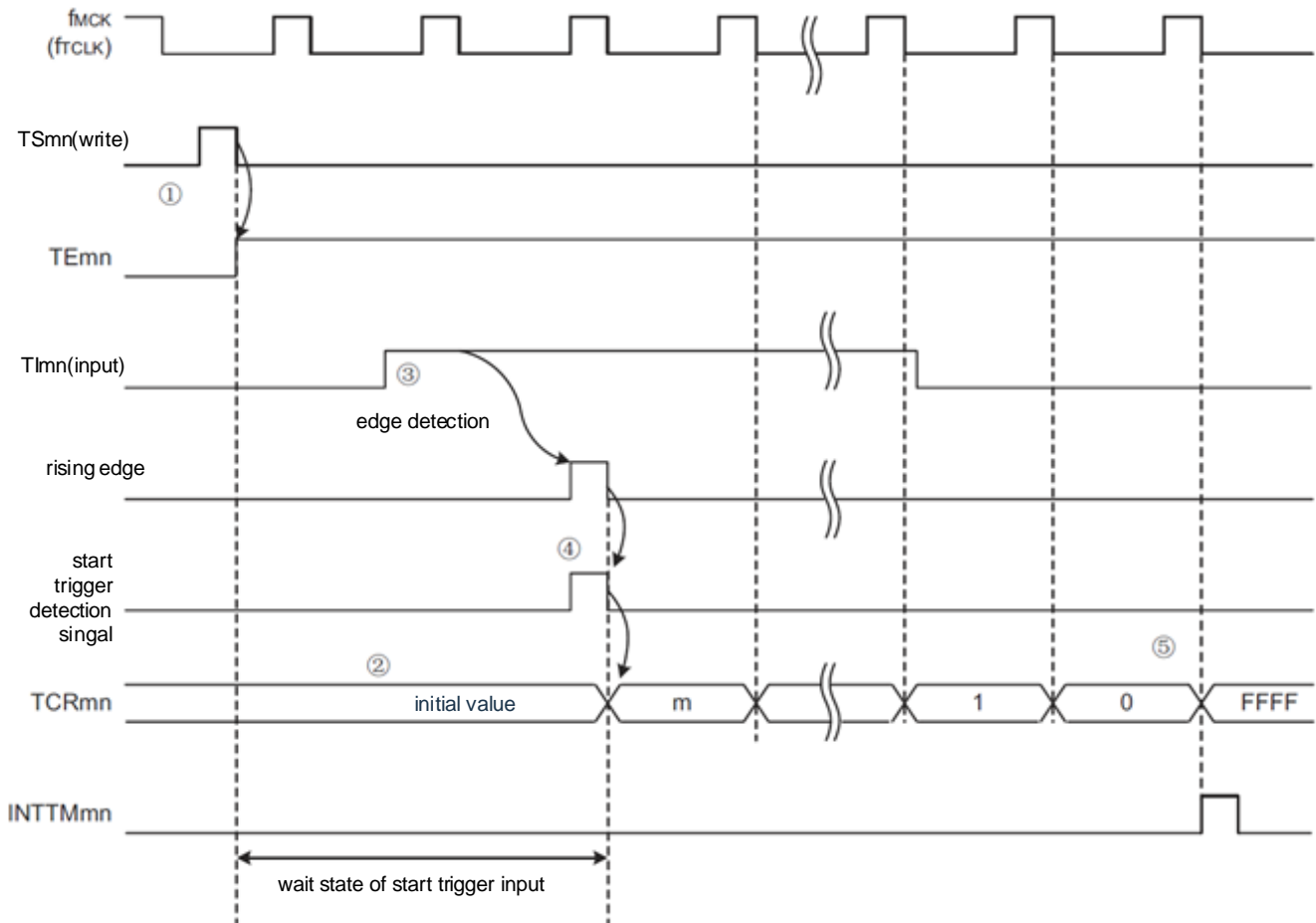
information about the start of the count timing, set MDmn0 to “1” so that an interrupt can be generated at the start of the count.

Note 2: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TImn input. The 1 cycle error is because the TImn input is not synchronized with the count clock (F_{MCK}).

(4) Operation of single count mode

- ① The operation enable state is entered by writing "1" to the TSmn bit (TEmn=1).
- ② The timer count register mn (TCRmn) remains the initial value until a start trigger signal is generated.
- ③ Detects the rising edge of the TImn input.
- ④ The value (m) of the TDRmn register is loaded into the TCRmn register after a start trigger signal is generated, and counting begins.
- ⑤ When the TCRmn register decrements to "0000H", the INTTMmn interrupt is generated and the value of TCRmn register changes to "FFFFH" and stop counting.

Figure 6-7: Operation timing (single count mode)

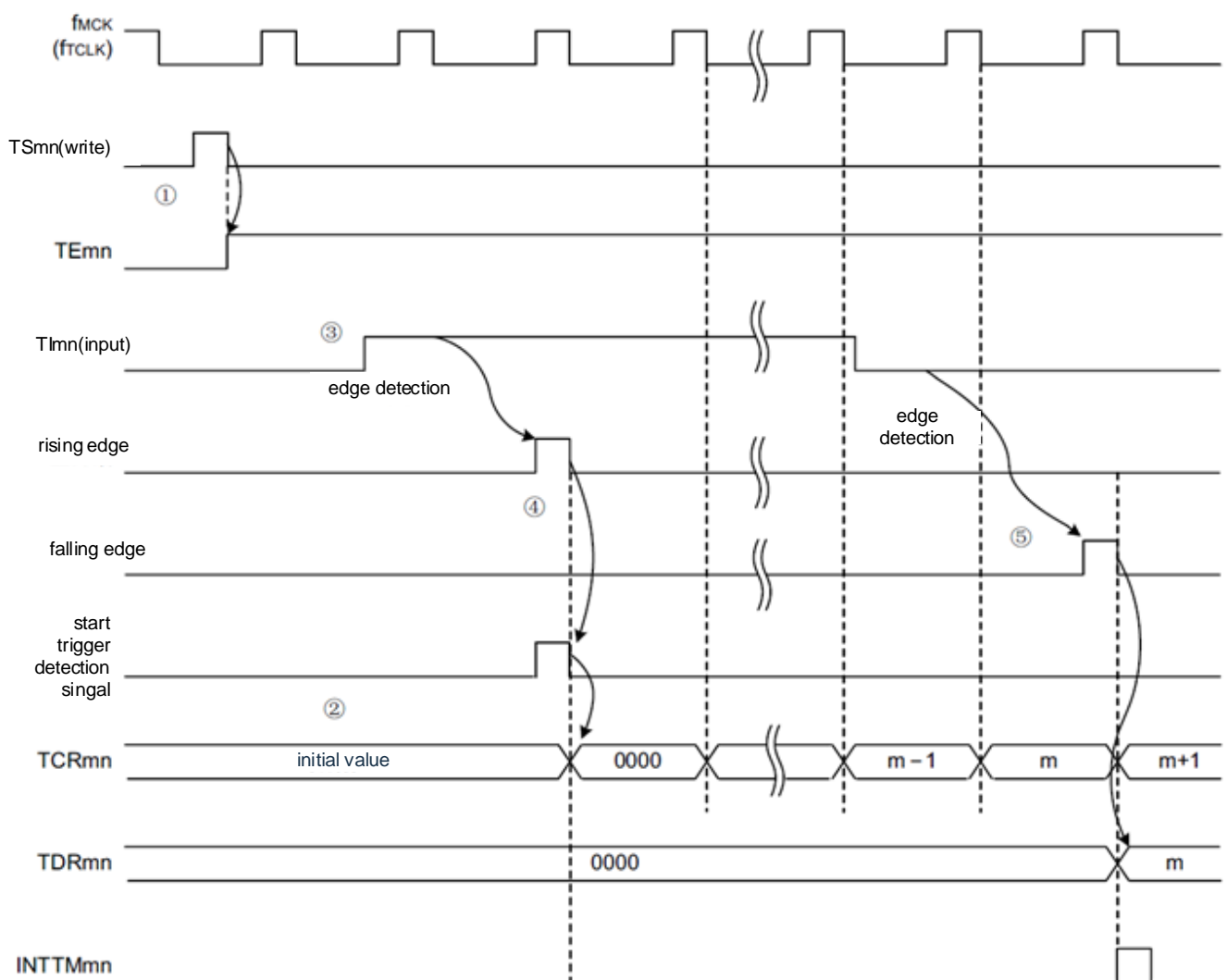


Note 1: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TImn input. The 1 cycle error is because the TImn input is not synchronized with the count clock (F_{MCK}).

(5) Operation of capture & single count mode (measurement of high-level width)

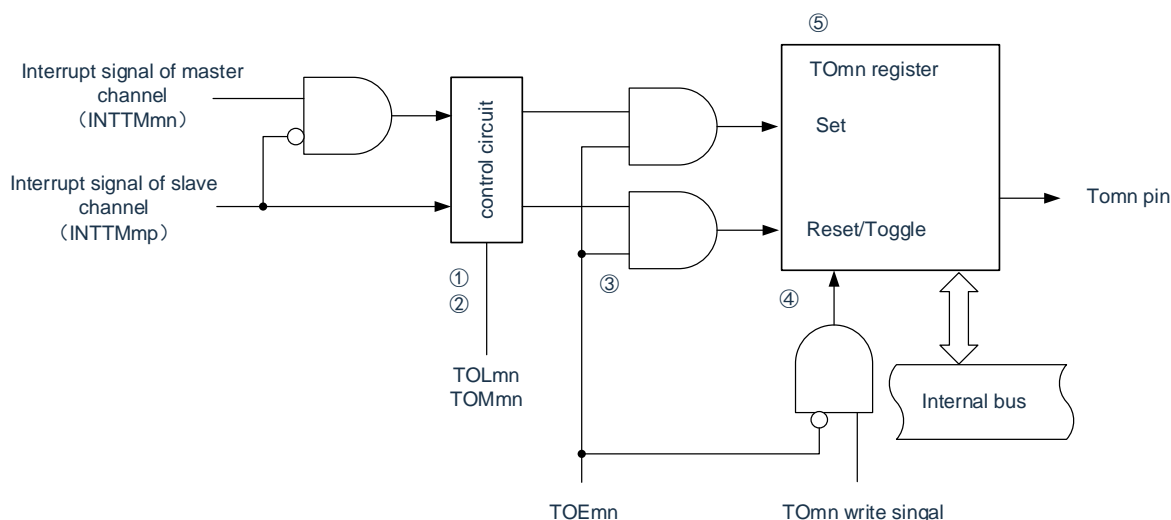
- ① The operation enable state is entered by writing "1" to the TSmn bit of the timer channel start register m (TSMn)(TEMn=1).
- ② The timer count register mn (TCRmn) remains the initial value until a start trigger signal is generated.
- ③ Detects the rising edge of the TImn input.
- ④ After the start trigger signal is generated, "0000H" is loaded into the TCRmn register and counting starts.
- ⑤ If the falling edge of TImn input is detected, the value of the TCRmn register is captured to the TDRmn register and an INTTMmn interrupt is generated.

Figure 6-8: Operation timing (capture & single count mode: measurement of high-level width)



Note: This is a timing without the noise filter. If the noise filter is used, the edge detection is delayed by 2 more F_{MCK} cycles (3~4 cycles in total) from the TImn input. The 1 cycle error is because the TImn input is not synchronized with the count clock (F_{MCK}).

6.5.1 T0mn pin output circuit configuration

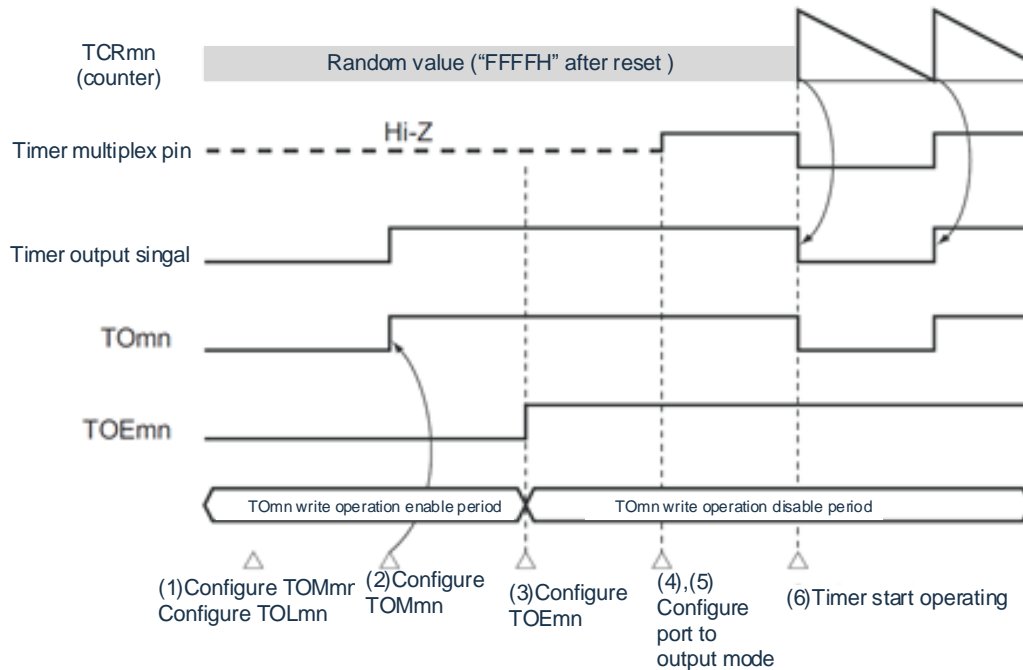


- ① When the TOMmn bit is “0” (master channel output mode), the setting value of timer output level register m (TOLm) is ignored and only INTTMmp (slave channel timer interrupt) is passed to timer output register m (TOM).
- ② When the TOMmn bit is “1” (slave channel output mode), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are passed to the TOM register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:
When TOLmn = 0: Positive logic output (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTMmp → set)
When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.
- ③ In the state of enabling timer output (TOEmn=1), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are passed to TOM register. Writing to the TOM register (TOMn write signal) is invalid.
When the TOEmn bit is “1”, the output of the TOMn pin is not changed except for the interrupt signal. To initialize the output level of the TOMn pin, you need to write a value to the TOM register after setting it to disable the timer output (TOEmn=0).
- ④ Writing to the TOMn bit for the object channel (TOMn write signal) is valid when the timer output is disabled (TOEmn=0). When the timer output is disabled (TOEmn=0), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are not passed to the TOM register.
- ⑤ The TOM register can be read at any time and the output level of the TOMn pin can be confirmed.

6.5.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6-10: State change from setting timer output to start of operation



- ① Set the operation mode of the timer output.
TOMmn bit (0: master channel output mode, 1: slave channel output mode)
TOLmn bit (0: positive logic output, 1: negative logic output)
- ② The timer output signal is set to the initial state by setting the timer output register m (TOM).
- ③ Writing "1" to TOEmn bit enables timer output (writing to TOM register is disabled).
- ④ The port is set to digital input/output via the port mode control register (PMCxx)
- ⑤ Set the input/output of the port to output
- ⑥ Enable timer operation (TSmn=1).

Note: m: unit number (m=0) n: channel number (n=0~3)

6.5.3 Cautions on channel output operation

- (1) Change of setting values for TOM, TOEm, TOLm, TOMm registers in timer operation

The operation of the timer (timer count register mn (TCRmn) and timer data register mn (TDRmn)) and the Tomn output circuit are independent. Therefore, changes in the setting values of timer output register m (TOM), timer output enable register m (TOEm), and timer output level register m (TOLm) do not affect the operation of the timer, and the setting values can be changed during timer operation. However, in order to output the expected waveform from the TOMn pin during the operation of each timer, the value must be set to the example of the register setting contents for each operation shown in Sections 6.7 and 6.8.

If the setting values of TOEm register and TOLm register other than TOM register are changed before and after generating the timer interrupt (INTTMmn) signal for each channel, the waveform output from TOMn pin may be different depending on whether it is changed before or after generating the timer interrupt (INTTMmn) signal.

Note: m: unit number (m=0) n: channel number (n=0~3)

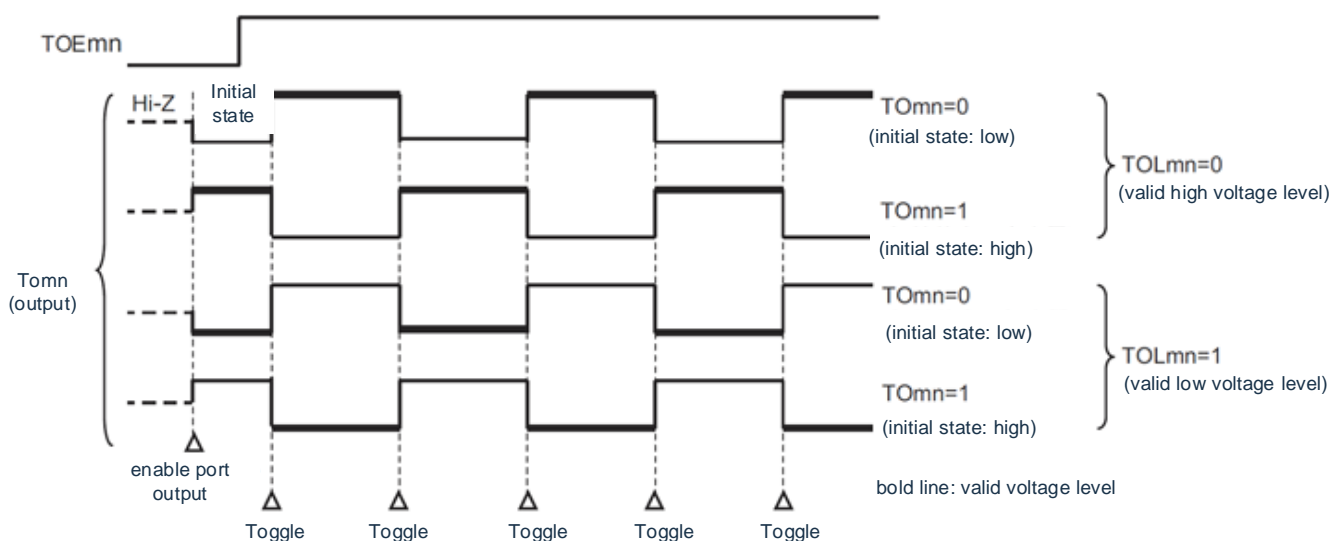
- (2) Default level of TOMn pin and output level after timer operation start

The change in the output level of the TOMn pin when timer output register m (TOM) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

- 1) Operation starts in master channel output mode (TOMmn=0)

In the master channel output mode (TOMmn=0), the setting of the timer output level register m (TOLm) is invalid. If the timer operation is started after the initial level is set, the output level of the TOMn pin is inverted by generating a toggle signal.

Figure 6-11: Output state of TOMn pin at toggle output (TOMmn=0)



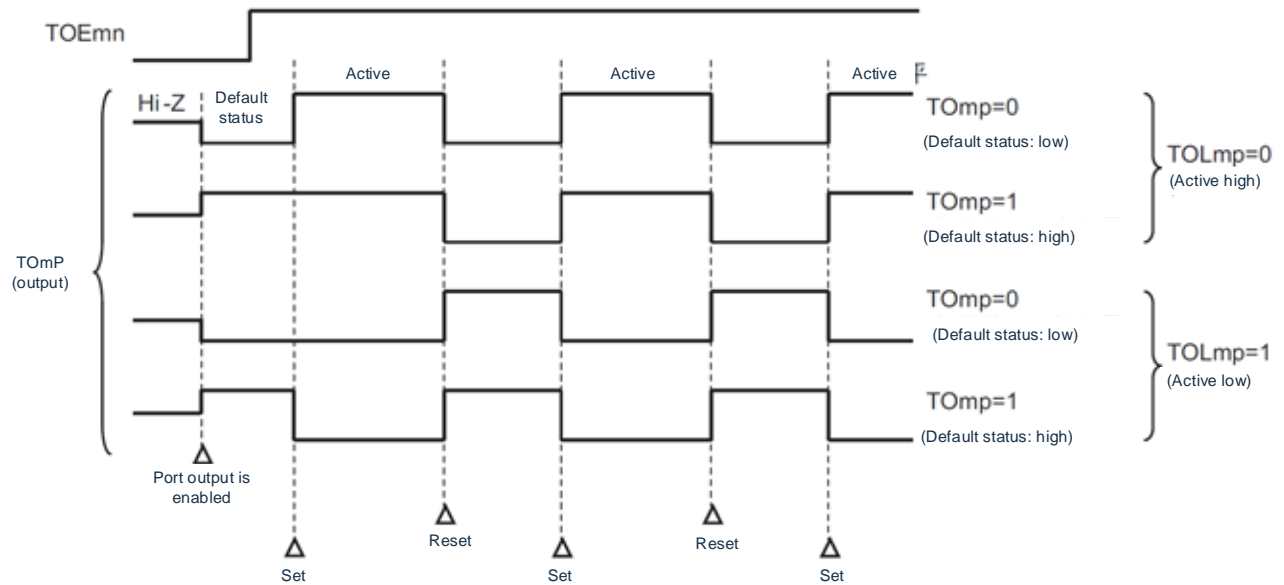
Note 1: Toggle: Reverse TOMn pin output status.

Note 2: m: unit number (m=0) n: channel number (n=0~3)

1) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))

In slave channel output mode (TOMmn=1), the active level depends on the setting of timer output level register m (TOLmn).

Figure 6-12: Output state of TOmn pin at PWM output (TOMmn=1)



Note 1: Set: The output signal from the TOmp pin changes from an invalid level to a valid level.

Note 2: Reset: The output signal from the TOmp pin changes from a valid level to an invalid level.

Note 3: m: unit number (m=0) n: channel number (p=1~3)

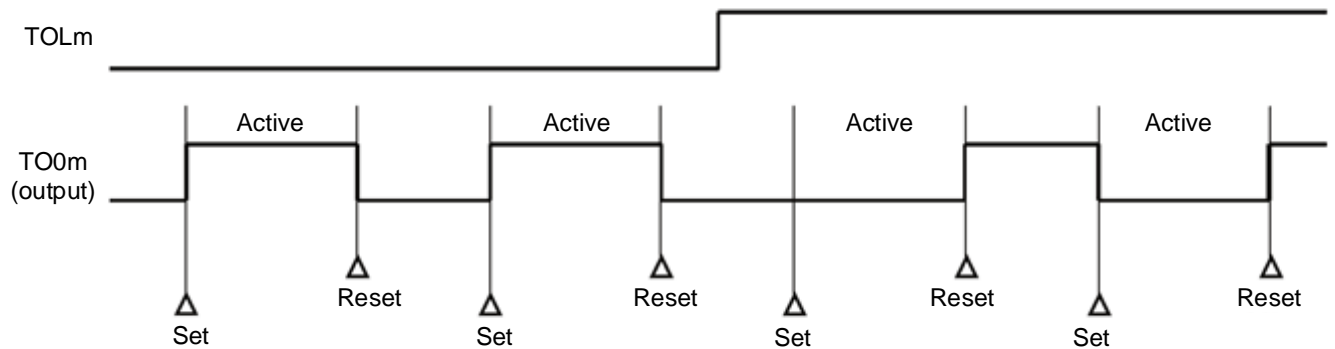
(3) Operation of TOMn pin in slave channel output mode (TOMmn = 1)

1) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOMn pin change condition. Rewriting the TOLm register does not change the output level of the TOMn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEMn = 1) is shown below.

Figure 6-13: Operation when the contents of the TOLm register are changed during timer operation



Note 1: The output signal from the TOMn pin changes from an invalid level to a valid level.

Note 2: Reset: The output signal from the TOMn pin changes from a valid level to an invalid level.

Note 3: m: unit number (m=0) n: channel number (n=0~3)

2) Set/reset timing

3) In order to achieve 0% and 100% output at PWM output, the set timing of the TOMn pin/TOMn bit when generating the master channel timer interrupt (INTTMmn) is delayed by 1 count clock via the slave channel.

When the set condition and reset condition are generated at the same time, the reset condition is given priority.

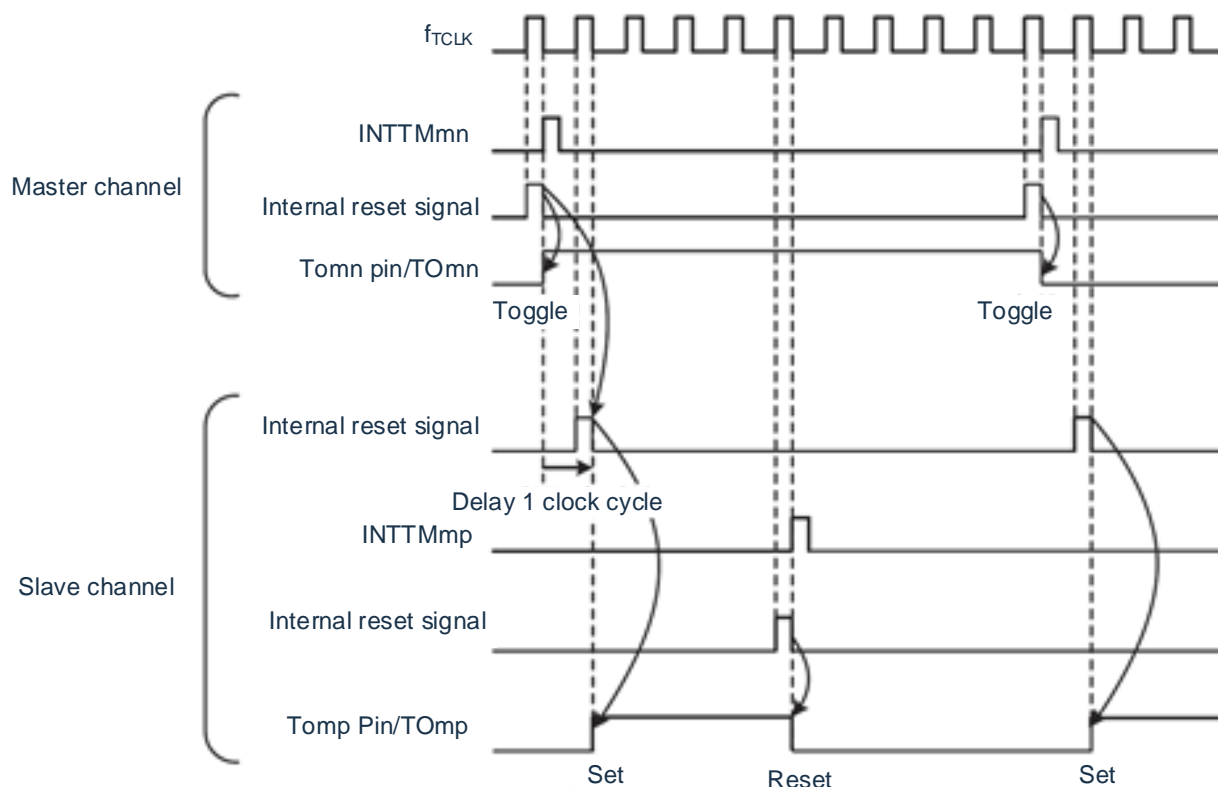
The set/reset operation status when setting the master/slave channel according to the following method is shown in Figure 6-14.

Master channel: TOEmn=1, TOMmn=0, TOLmn=0

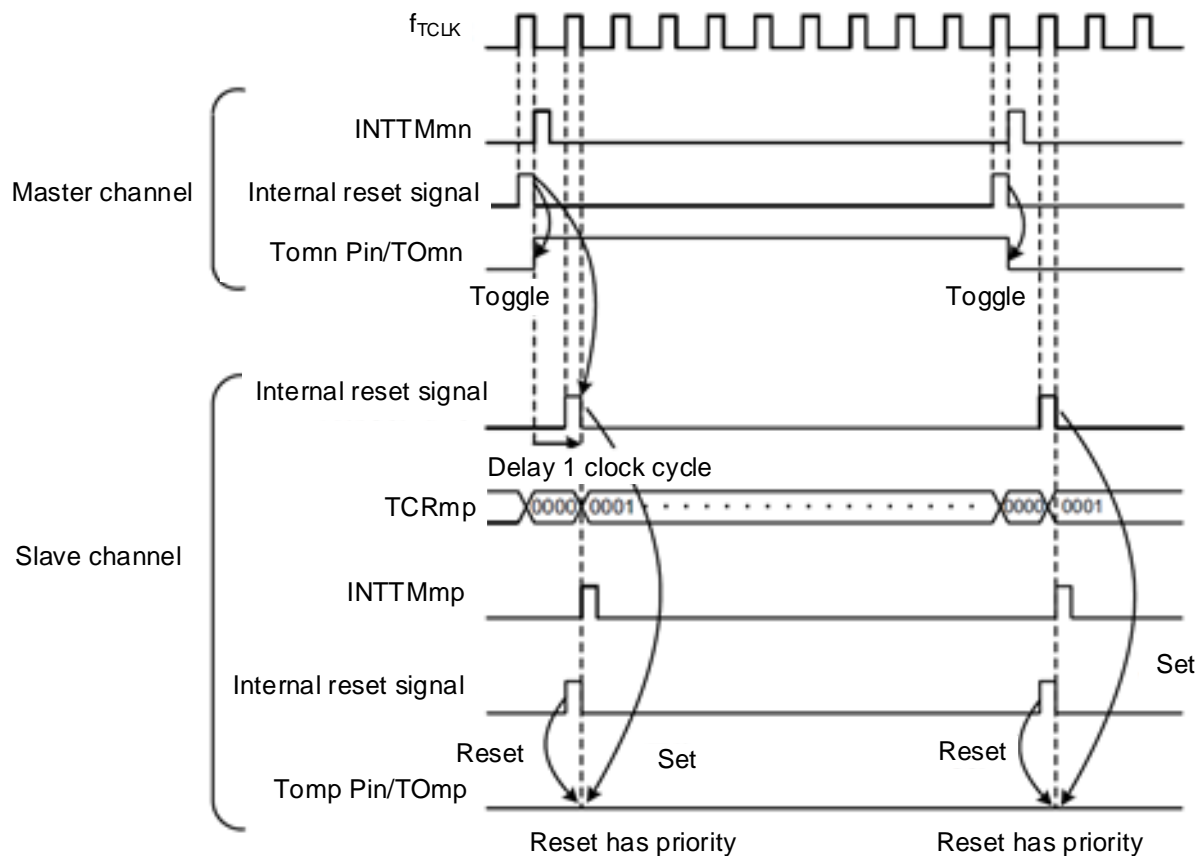
Slave channel: TOEmp=1, TOMmp=1, TOLmp=0

Figure 6-14: Set/reset timing operation status

(1) Basic operation timing



(2) Operation timing when 0% duty cycle



Note 1: Internal reset signal: TOMn pin reset/toggle signal

Note 2: Internal set signal: TOn pin set signal

Note 3: m: unit number (m=0) n: channel number n=0~3 (master channel: n=0, 2)

p: slave channel number n=0: p=1, 2, 3 n=2: p=3

6.5.4 One-time operation of TOn bit

Like the timer channel start register m (TSm), the timer output register m (TOn) has the set bits (TOn) for all channels and can therefore operate the TOn bits for all channels at once.

Table 6-27: One-time operation example of TOn bit

Before writing

| | | | | | | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|---|------|------|------|------|
| TO0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TO03 | TO02 | TO01 | TO00 |
| | | | | | | | | | | | | 1 | 0 | 1 | 0 |

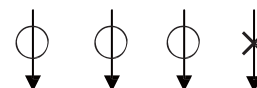
| | | | | | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|---|---|-------|-------|-------|-------|
| TOE0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TOE03 | TOE02 | TOE01 | TOE00 |
| | | | | | | | | | | | | 0 | 0 | 0 | 1 |

Data to be written

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

After writing

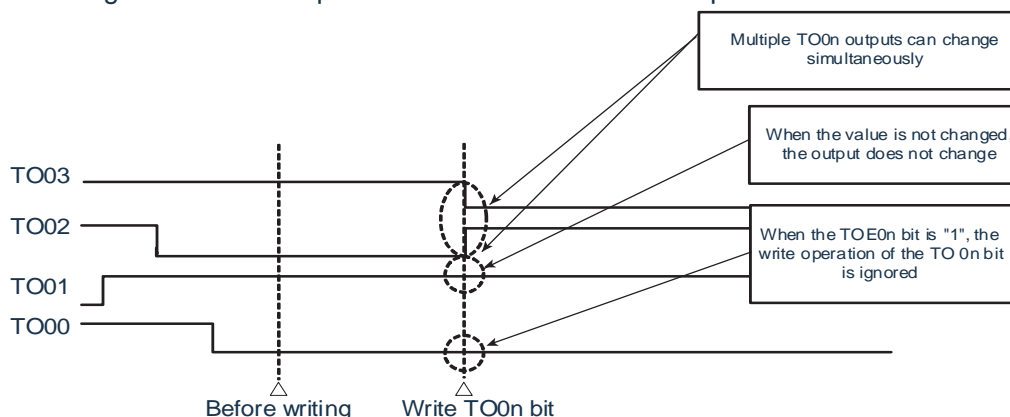
| | | | | | | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|---|------|------|------|------|
| TO0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TO03 | TO02 | TO01 | TO00 |
| | | | | | | | | | | | | 0 | 1 | 1 | 0 |



Only TOn bits with TOE0n bit "0" can be written, and write is ignored when the TOn bit is "1".

TOn (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TOn bit, it is ignored and the output change by timer operation is normally done.

Figure 6-15: TOn pin state when the TOn bit is operated at one time



Note: m: unit number (m=0) n: channel number (n=0~3)

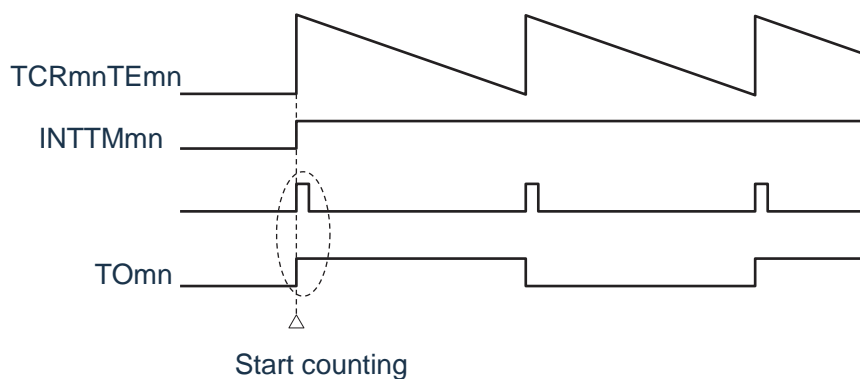
6.5.5 Timer interrupt and TOmn pin output when counting starts

In interval timer mode or capture mode, the MDmn0 bit of timer mode register mn (TMRmn) is the bit that sets whether to generate a timer interrupt when counting starts.

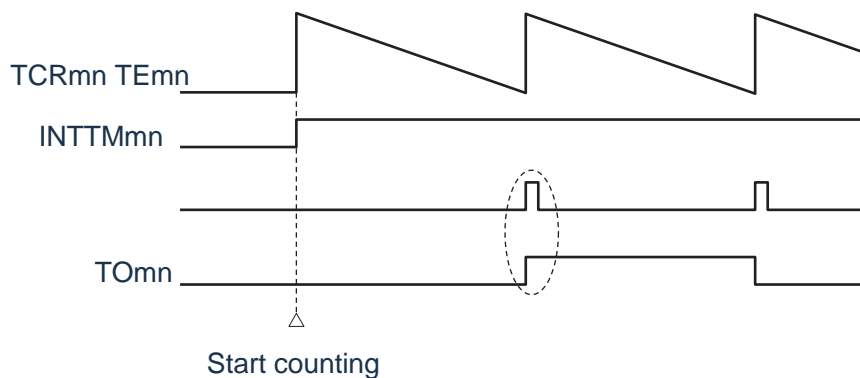
When the MDmn0 bit is “1”, the start timing of the count can be known by generating a timer interrupt (INTTMmn). In other modes, the timer interrupt and TOmn output at the start of counting are not controlled. An example of operation when set to interval timer mode (TOEmn=1, TOMmn=0) is shown below.

Figure 6-16: An operation example of timer interrupt and TOmn output at start count

(a) When MDmn0 = 1



(b) When MDmn0 = 0



When MDmn0 bit is “1”, the timer interrupt (INTTMmn) is output at the start of counting and TOmn is output alternately.

When MDmn0 bit is “0”, no timer interrupt (INTTMmn) is output at the start of counting and TOmn is not changed, while INTTMmn is output and TOmn is alternately output after 1 cycle of counting.

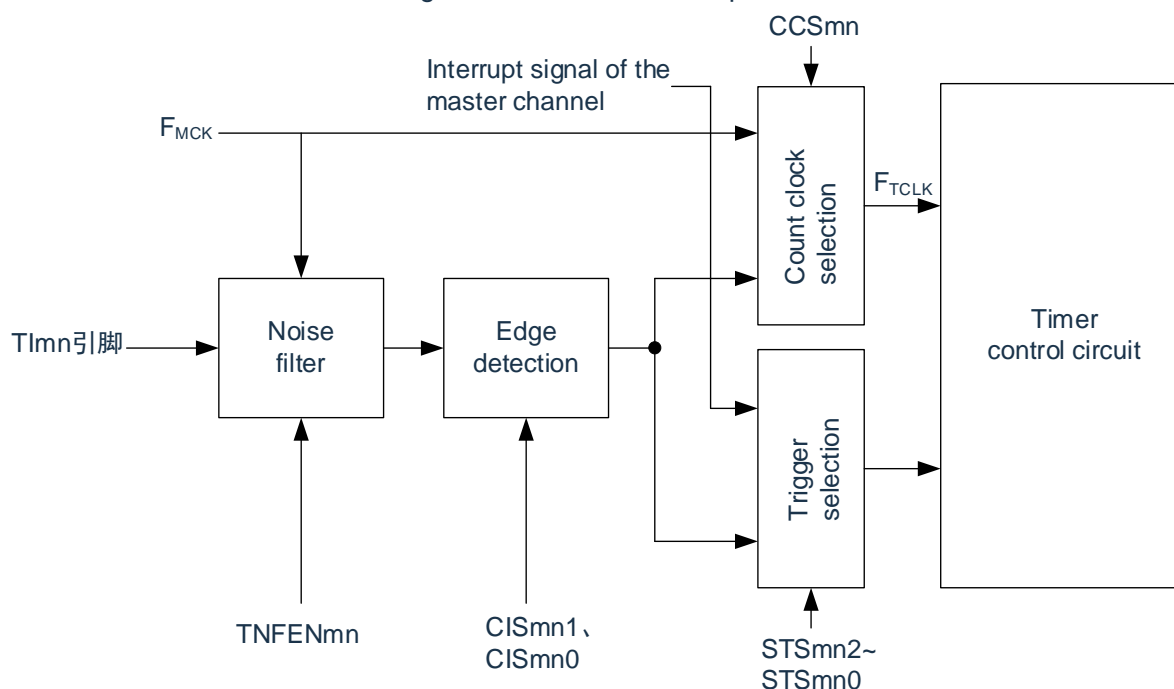
Note: m: unit number (m=0) n: channel number (n=0~3)

6.6 Control of timer input (Tlmn)

6.6.1 Structure of Tlmn pin input circuit

The signal from the timer input pins is input to the timer control circuit via a noise filter and the edge detection circuit. For pins that need to be removed from noise, the corresponding pin noise filter must be set to enable. The block diagram of the input circuit is as follows.

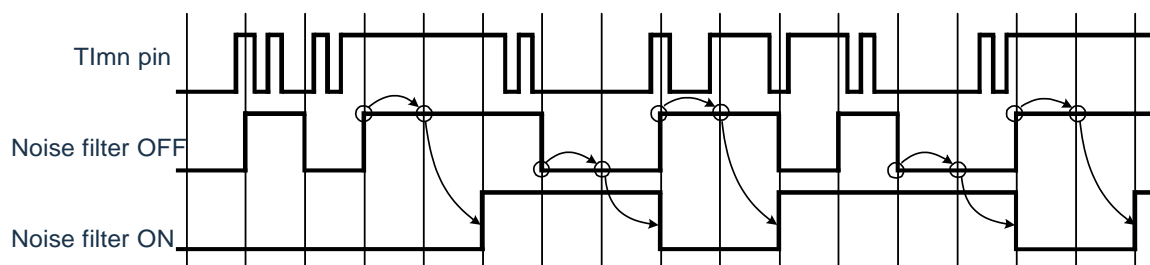
Figure 6-17: Structure of input circuit



6.6.2 Noise filter

When the noise filter is inactive, synchronization is performed only by the operation clock (F_{MCK}) of channel n. When the noise filter is active, 2 clocks are detected after synchronization by the operation clock (F_{MCK}) of channel n. The waveform of the TM4mn input pin after the noise filter circuit with the noise filter ON or OFF is shown below.

Figure 6-18: Sample waveform of Tlmn input pin with noise filter ON or OFF



Note: The input waveform on the Tlmn pin is used to illustrate the operation of the noise filter is ON or OFF. For actual operation, the input must be made in accordance with the Tlmn input high- and low-level width shown in AC characteristics.

6.6.3 Cautions on channel input operation

When set to not use the timer input pin, no operating clock is provided to the noise filter circuit. Therefore, the following wait time is required from the time set to use the timer input pin to the time the channel corresponding to the timer input pin is set to operate the enable trigger.

(1) Noise filter OFF

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set, wait for at least two cycles of the operating clock (F_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter ON

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set, wait for at least four cycles of the operating clock (F_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

6.7 Independent channel operation function of general-purpose timer unit

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

It can be used as a reference timer to generate INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated using the following equation:

$$\text{INTTMmn (timer interrupt) generation period} = \text{count clock period} \times (\text{TDRmn set value} + 1)$$

(2) Operation as square wave output

The TOMn alternates outputs while generating the INTTMmn, outputting a square wave with a 50% duty cycle.

The period and frequency of the square wave can be calculated using the following equation:

$$\text{Period of square wave output from TOMn} = \text{Period of count clock} \times (\text{TDRmn set value} + 1) \times 2$$

$$\text{Frequency of square wave output from TOMn} = \text{Frequency of count clock} / \{(\text{TDRmn set value} + 1) \times 2\}$$

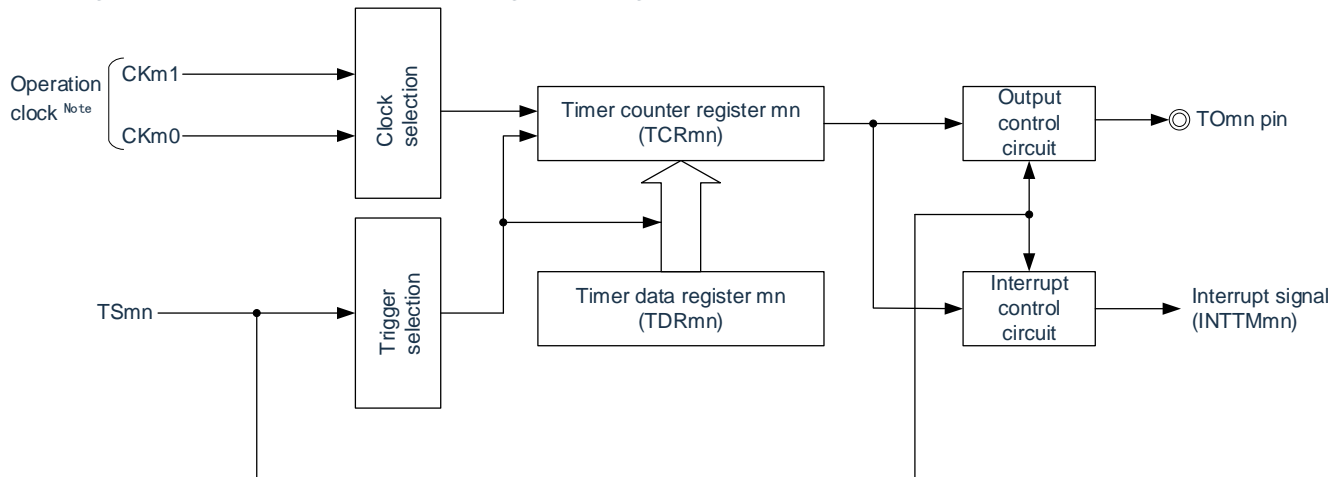
In the interval timer mode, the timer count register mn (TCRmn) is used as a decrement counter.

After setting the channel start trigger bit (TSmn, TSHm1, TSHm3) of the timer channel start register m (TSM) to "1", the value of timer data register mn (TDRmn) is loaded into the TCRmn register by the first count clock. At this time, if the MDmn0 bit of the timer mode register n (TMRmn) is "0", INTTMmn is not output and TOMn is not alternately output. If the MDmn0 bit of TMRmn register is "1", INTTMmn is output and TOMn is alternately output. Then, the TCRmn register is decremented by the count clock.

If the TCRmn becomes "0000H", the INTTMmn and TOMn are output alternately by the next count clock. At the same time, the value of TDRmn register is loaded into TCRmn register again. After that, continue the same operation.

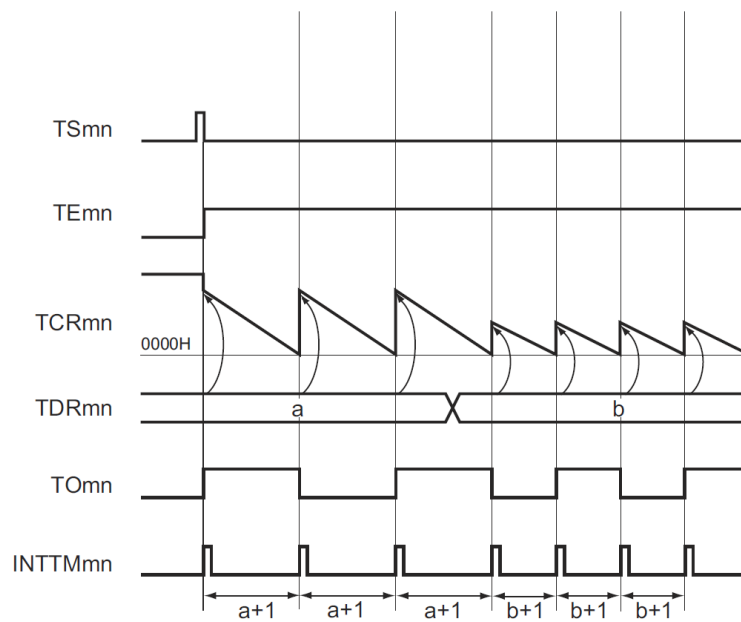
The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid from the next cycle.

Figure 6-19: Example of basic timing operating as an interval timer/square wave output (MDmn0=1)



Note: At channel 1 and channel 3, it is possible to select the clock from CKm0, CKm1, CKm2 and CKm3.

Figure 6-20: Example of basic timing operating as an interval timer/square wave output (MDmn0=1)



Note 1: m: unit number (m=0) n: channel number (n=0~3)

Note 2: TSmn: Bit n of timer channel start register m (TSM)

TEmn: Bit n of timer channel enable status register m (TEM)

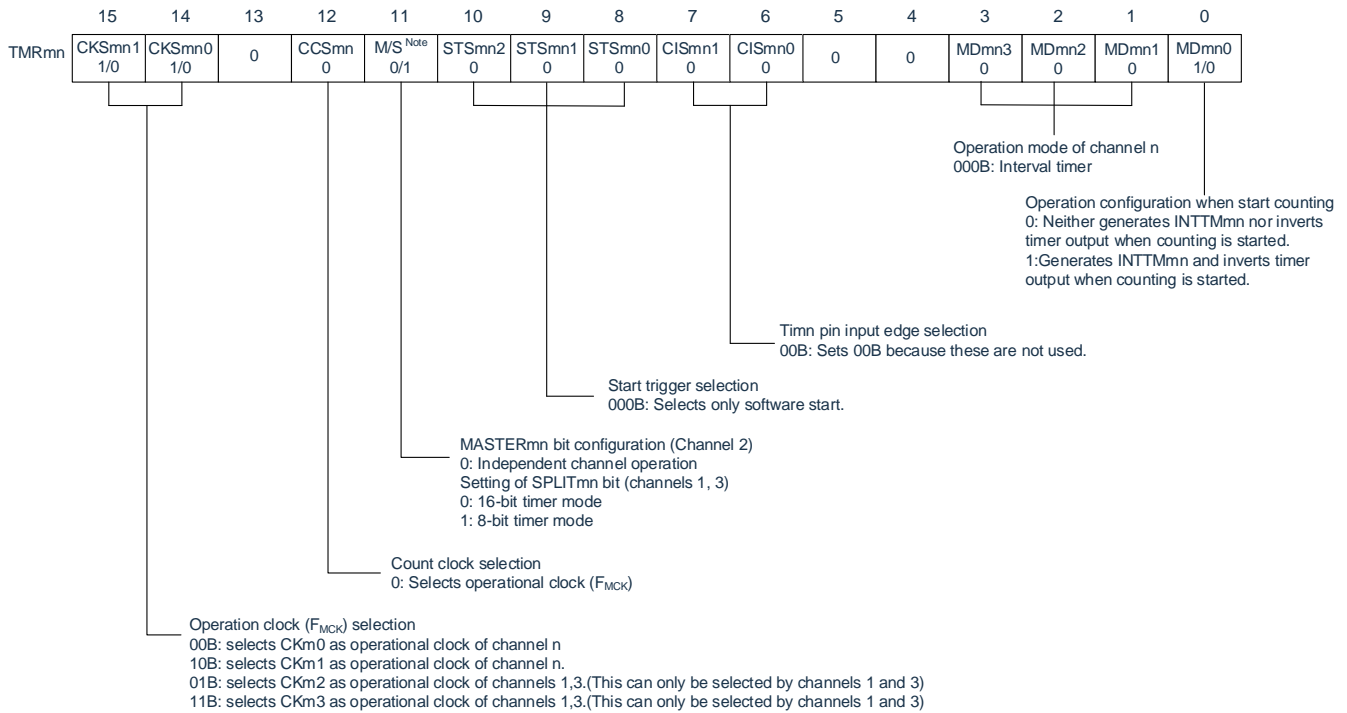
TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

TOMn: TOMn pin output signal

Figure 6-21: Example of register setting contents for interval timer/square wave output

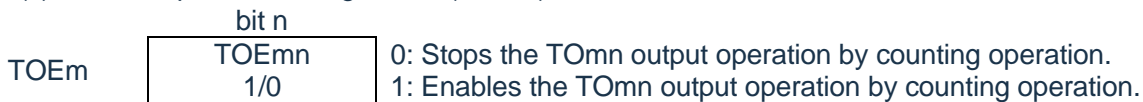
(a) Timer mode register mn (TMRmn)



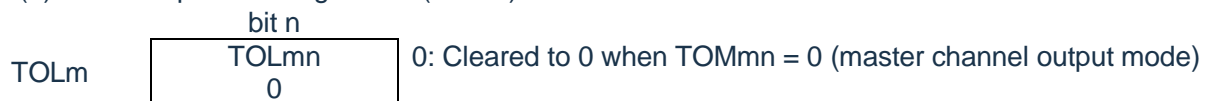
(b) Timer output enable register m (TOEm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmn bit

Note 2: TMRm1, TMRm3: SPLITmn bit

Note 3: TMRm0: Fixed to "0".

Note 4: m: unit number (m=0) n: channel number (n=0~3)

Table 6-28: Procedure for interval timer/square wave output function

| | Software operation | Hardware status |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAU initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the timer mode register mn (TMRmn) (to determine the channel's operation mode). Set the interval (period) value for the timer data register mn (TDRmn). | The channel is in the stop state. (Provides clock, and consumes some power) |
| | Using TOMn output: Set the TOMmn bit of Timer Output Mode Register m (TOMm) to "0" (master channel output mode). Set the TOLmn bit to "0". Set the TOMn bit to determine the initial level of the TOMn output. | The TOMn pin is in Hi-Z output state. |
| | Set the TOEmn bit to "1" and enable TOMn output. | When the port mode register is in output mode and the port register is "0", the TOMn initial set level is output. The TOMn remains unchanged because the channel is in the stop state. |
| | Set the Port Register and Port Mode Register to "0". | The TOMn pin outputs the level set by the TOMn. |
| Start operate | (The TOEmn bit set to "1" only when the TOMn output is used and restarted) Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0". | The TEMn bit becomes "1" and starts counting. Load the value of the TDRmn register into the Timer Count Register mn (TCRmn). When the MDmn0 bit of TMRmn register is "1", INTTMmn is generated and TOMn is output alternately. |
| Restart the operation | In operation The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used. The TOM register and TOEm register settings can be changed. The setting of the TMRmn register, the TOMmn bit and the TOLmn bit cannot be changed. | The counter (TCRmn) performs decremental counting. If the count reaches "0000H", the value of the TDRmn register is loaded into the TCRmn register again and the count continues. When TCRmn is detected as "0000H", INTTMmn is generated and TOMn is alternately output. Thereafter, repeat this operation. |
| | Stop operation Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit. | The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. |
| | | The TOMn output is not initialized but remains its state. |
| | Set the TOEmn bit to "0" and set the value for the TOMn bit. | The TOMn pin outputs the level set by the TOMn bit. |
| TAU stop | To maintain the output level of the TOMn pin: Set TOMn bit to "0" after setting the value to be held for the port register. No need to maintain the output level of the TOMn pin: No need to set. | The output level of the TOMn pin is maintained by the port function. |
| | Set the TM4mEN bit of the PER0 register to "0". | The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TOMn bit becomes "0" and TOMn pin becomes port function) |

Note: m: unit number (m=0) n: channel number (n=0~3)

6.7.2 Operation as external event counter

It can be used as an event counter to count the active edges (external events) detected on the TImn pin input and generate an interrupt if the specified count value is reached. The specified count value can be calculated using the following equation:

$$\text{Specified count value} = \text{TDRmn set value} + 1$$

In the event counter mode, the timer count register mn (TCRmn) is used as a decrement counter.

The value of timer data register mn (TDRmn) is loaded into the TCRmn register by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to “1”.

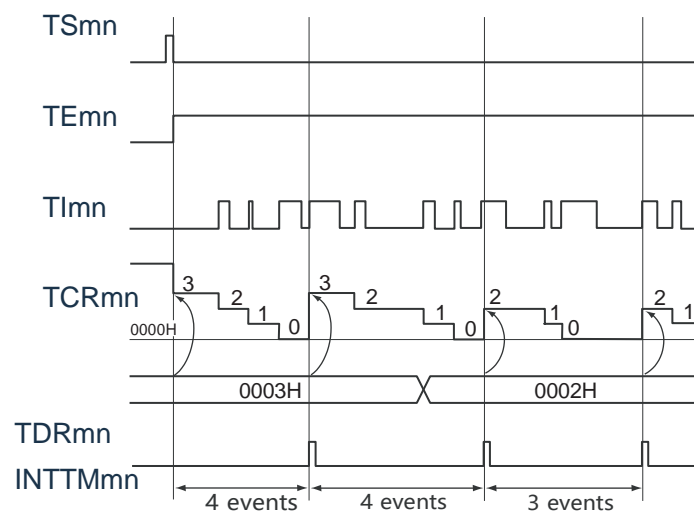
The TCRmn register decrements the count while detecting the active edge of the TImn pin input. If TCRmn becomes “0000H”, the value of TDRmn register is loaded again and INTTMmn is output.

After that, continue the same operation.

The output must be stopped by setting the TOEmn bit of the timer output enable register m (TOEm) to “0” because the TOMn pin outputs irregular waveforms based on external events.

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid for the next cycle.

Figure 6-22: Example of basic timing operating as external event counter



Note 1: m: unit number (m=0) n: channel number (n=0~3)

Note 2: TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

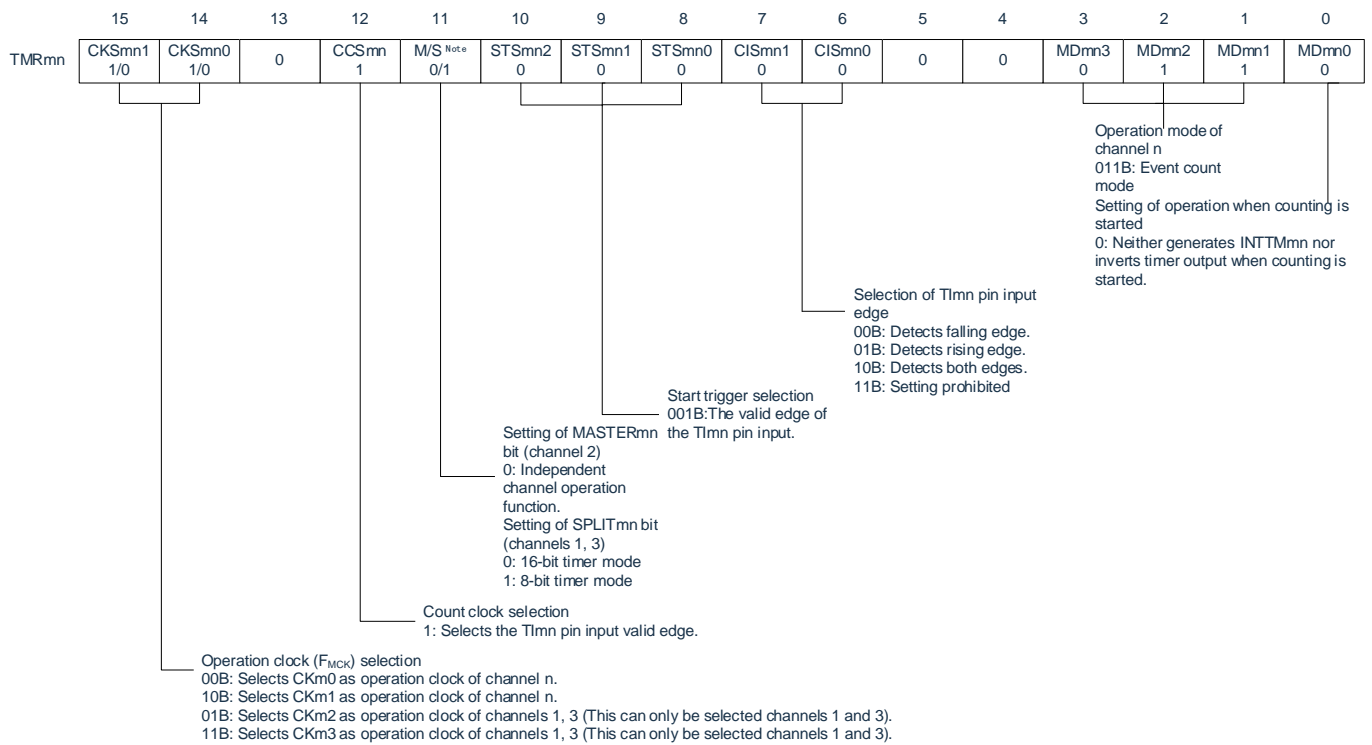
TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

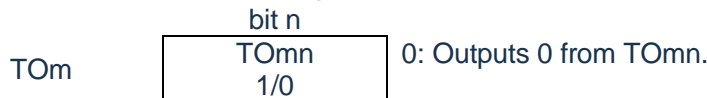
TDRmn: Timer data register mn (TDRmn)

Figure 6-23: Example of register contents setting in external event counter mode

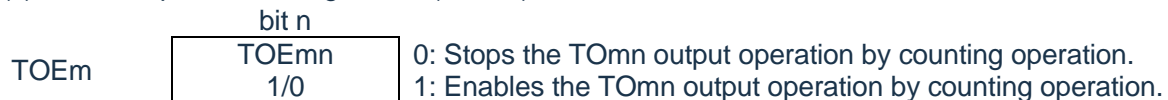
(a) Timer mode register mn (TMRmn)



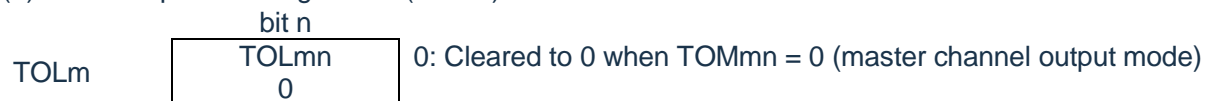
(b) Timer output enable register m (TOEm)



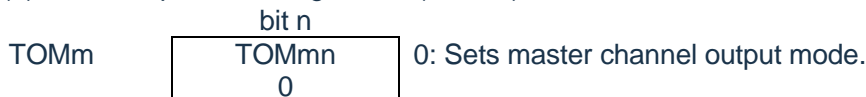
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmn bit

Note 2: TMRm1, TMRm3: SPLITmn bit

Note 3: TMRm0: Fixed to "0".

Note 4: m: unit number (m= 0) n: channel number (n=0~3)

Table 6-29: Procedure for external event counter function

| | Software operation | Hardware status |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). Set the count value for the timer data register mn (TDRmn). Set the TOEmn bit of the timer output enable register m (TOEm) to "0". | The channel is in the stop state. (Provides clock, and consumes some power) |
| Start operating | Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0". | The TEMn bit becomes "1" and starts counting. The value of the TDRmn register is loaded into the timer count register mn (TCRmn) and enter the detection wait state of the input edge of the TImn pin. |
| In operation | The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used. The setting of the TMRmn register, the TOMmn bit, the TOLmn bit, the Tomn bit and the TOEmn bit cannot be changed. | Whenever the input edge of the TImn pin is detected, the counter (TCRmn) is decremented. If the count reaches "0000H", the value of the TDRmn register is loaded into the TCRmn register again and the count continues. When TCRmn is detected as "0000H", INTTMmn is generated. Thereafter, repeat this operation. |
| Stop operating | Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit. | The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. |
| Timer4 stop | Set the TA4mEN bit of the PER0 register to "0". | The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. |

Restart operation

6.7.3 Operation as frequency divider

The clock input from the TI00 pin can be divided and used as a divider for the output of the TO00 pin. The divided clock frequency of the TO00 output can be calculated using the following equation:

- Select rising or falling edge:

$$\text{Divider clock frequency} = \text{input clock frequency} / \{(\text{TDR00 set value} + 1) \times 2\}$$
- Select both edges:

$$\text{Divider clock frequency} \approx \text{input clock frequency} / (\text{TDR00 set value} + 1)$$

In the interval timer mode, the timer count register 00 (TCR00) is used as a decrement counter.

After setting the channel start trigger bit (TS00) of timer channel start register 0 (TS0) to “1”, the value of timer data register 00 (TDR00) is loaded into the TCR00 register by detecting an active edge of TI00. At this time, if the MD000 bit of Timer Mode Register 00 (TMR00) is “0”, INTTM00 is not output and TO00 is not output alternately; if the MD000 bit of TMR00 register is “1”, INTTM00 is output and TO00 is not output alternately. If the MD000 bit of TMR00 register is “1”, INTTM00 is output and TO00 is output alternately.

The TCR00 register then counts down through the active edge of the TI00 pin input. If TCR00 changes to “0000H”, TO00 performs an alternate output. At the same time, the value of the TDR00 register is loaded into the TCR00 register and counting continues.

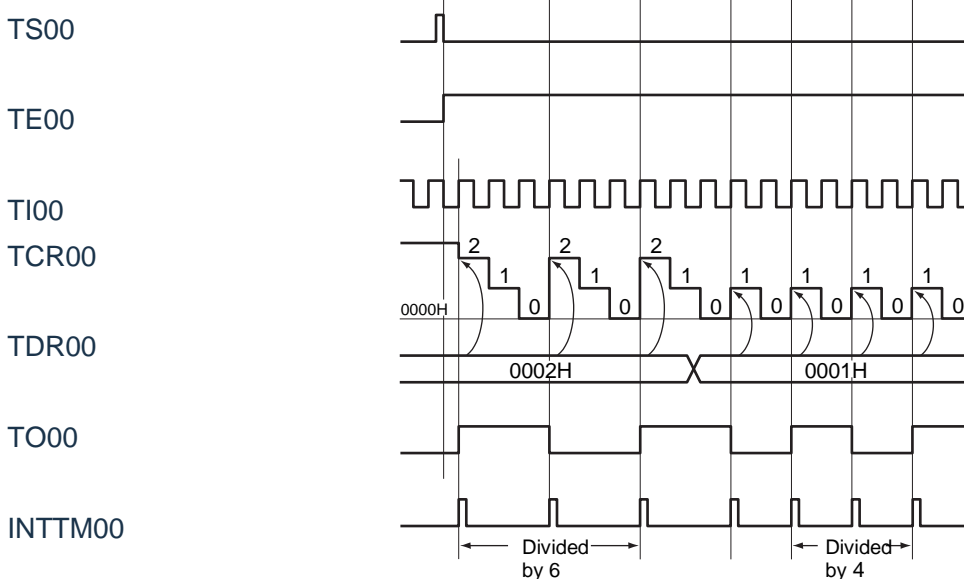
If double edge detection is selected for the TI00 pin input, the duty cycle error of the input clock affects the clock period of the TO00 output’s division.

The clock period of the TO00 output contains the sampling error of 1 run clock cycle.

$$\text{clock period of the TOmn output} = \text{supposed TOmn output clock period} \pm \text{operating clock period}$$

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid for the next cycle.

Figure 6-24: Example of basic timing operating as a frequency divider (MD000=1)



Note: TS00: Bit 0 of timer channel start register 0 (TS0)

TE00: Bit 0 of timer channel enable status register (TE0)

TI00: TI00 pin input signal

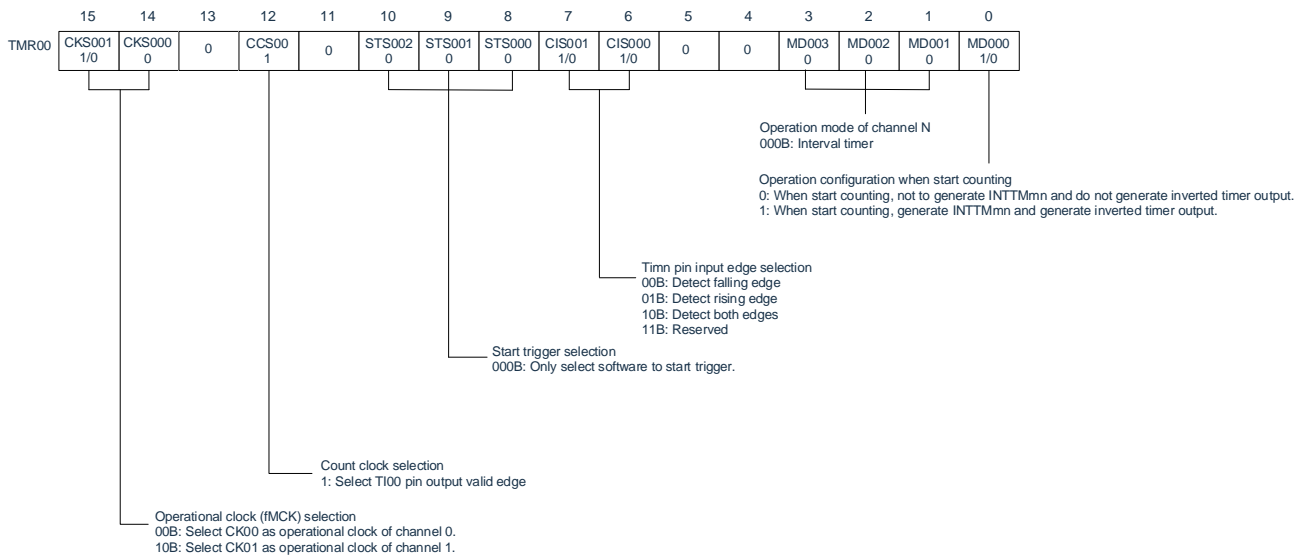
TCR00: Timer count register 00 (TCR00)

TDR00: Timer data register 00 (TDR00)

TO00: TO00 pin output signal

Figure 6-25: Example of register contents setting when operating as a frequency divider

(a) Timer mode register 00 (TMR00)



(b) Timer output register 0 (TO0)

| | | | |
|-----|-------|-----|----------------------------------------------------|
| | bit 0 | | |
| TO0 | TO00 | 1/0 | 0: Outputs "0" by TO00. 1: Outputs "1" by TO00. |
| | | | |

(c) Timer output enable register 0 (TOE0)

| | | | |
|------|-------|-----|------------------------------------------------------------------------------------------------------------------------------|
| | bit 0 | | |
| TOE0 | TOE00 | 1/0 | 0: Stops the TO00 output performed by the counting operation. 1: Enables TO00 output performed by the counting operation. |
| | | | |

(d) Timer output level register 0 (TOL0)

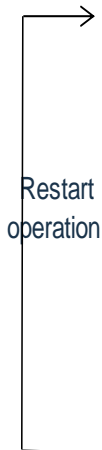
| | | | |
|------|-------|---|----------------------------------------------------------|
| | bit 0 | | |
| TOL0 | TOL00 | 0 | 0: Sets "0" in the master channel output mode (TOM00=0). |
| | | | |

(e) Timer output mode register 0 (TOM0)

| | | | |
|------|-------|---|-------------------------------------|
| | bit n | | |
| TOM0 | TOM00 | 0 | 0: Sets master channel output mode. |
| | | | |

Table 6-30: Procedure for frequency divider function

| | Software operation | Hardware status |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit 0 is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of peripheral enable register 0(PER0) to "1". | The input clock of timer unit 0 is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register 0 (TPS0). Determine the clock frequency of CK00 ~ CK03. | |
| Initial setting of channels | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register 00 (TMR00)(to determines the operating mode of the channel, select edge detection). Set the interval (cycles) value for the timer data register 00 (TDR00). | The channel is in the stop state. (Provides clock, and consumes some power) |
| | Set TOM00 bit of timer output mode register 0 (TOM0) to "0" (master control channel output mode). The TOL00 bit must be set to "0". Set the TO00 bit and determine the initial level of TO00 output | The TO00 pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TO00 initial set level is output. Since the channel is in stop state, TO00 does not change. The TO00 pin outputs the level set by the TO00. |
| | Set TOE00 bit to "1" and enable TO00 output. | |
| | Set the Port Register and Port Mode Register to "0" | |
| | | |
| Start operation | Set TOE00 bit to "1" (only limited to restart operation). The TS00 bit must be set to "1". The operation automatically returns to "0" because the TS00 bit is a trigger bit. | The TE00 bit becomes "1" and starts counting. Load the value of the TDR00 register into the Timer Count Register 00 (TCR00). When the MD000 bit TMR00 register is "1", INTTM00 is generated and TO00 is output alternately. |
| | The setting of the TDR00 register can be changed at will. The TCR00 register can be read at any time. The TSR00 register is not used. The TO0 register and TOE0 register settings can be changed. The setting of the TMR00 register, the TOM00 bit and the TOL00 bit cannot be changed. | The counter (TCR00) performs decremental counting. If the count reaches "0000H", the value of the TDR00 register is loaded into the TCR00 register again and the count continues. When the TCR00 bit is "0000H", INTTM00 is generated and TO00 is output alternately. Thereafter, repeat this operation. |
| Stop operating | The TT00 bit must be set to "1". The operation automatically returns to "0" because the TT00 bit is a trigger bit. | The TE00 bit becomes "0" and starts counting. The TCR00 register holds the count value and stops counting. The TO00 output is not initialized but remains its state. |
| | Set the TOE00 bit to "0" and set the value for the TO00 bit. | The TO00 pin outputs the level set by the TO00. |
| Timer4 stop | To maintain the output level of the TO00 pin: Set TO00 bit to "0" after setting the value to be held for the port register. No need to maintain the output level of the TO00 pin: No need to set. | The output level of the TO00 pin is maintained by the port function. |
| | Set the TM4mEN bit of the PER0 register to "0". | The input clock of timer unit 0 is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TO00 bit becomes "0" and TO00 pin becomes port function) |



6.7.4 Operation as input pulse interval measurement

The count value can be captured at the active edge of TImn and the interval between TImn input pulses can be measured. The software operation (TSMn=1) can also be set to capture the count value during the period when the TEMn bit is “1”.

The pulse interval can be calculated using the following equation:

$$\text{TImn input pulse interval} = \text{period of counting clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{TDRmn captured value} + 1))$$

Note: The 1 operation clock error is generated because the TImn pin input is sampled by the operation clock selected by the CKSmn bit of the Timer Mode Register mn (TMRmn).

In capture mode, the timer count register mn (TCRmn) is used as an increment counter.

If the channel start trigger bit (TSMn) of the timer channel start register m (TSM) is set to “1”, the TCRmn register is incrementally counted from “0000H” by the count clock.

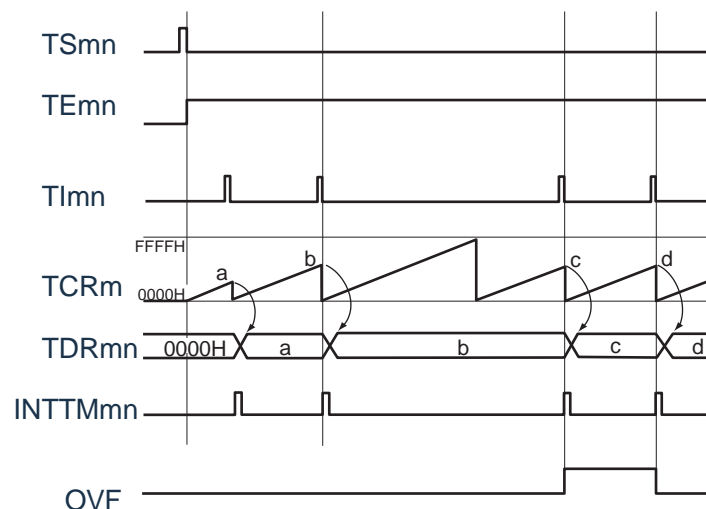
If the active edge of the TImn pin input is detected, the count value of TCRmn register is transferred (captured) to Timer Data Register mn (TDRmn), and the TCRmn register is cleared to “0000H”, and then INTTMmn is output. If the counter overflows, the OVF bit of Timer Status Register mn (TSRmn) is set to “1”. If the counter does not overflow, the OVF bit is cleared. After that, continue the same operation.

While capturing the count value to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether or not overflow occurs during the measurement, and the overflow status of the captured value can be confirmed.

Even if the counter counts 2 or more complete cycles, the overflow is considered to have occurred and the OVF bit of the TSRmn register is set to “1”. However, when two or more overflows occur, the interval value cannot be measured normally by the OVF bit.

Set the STSmn2~STSmn0 bit of the TMRmn register to “001B”, and use the valid edge of TImn for start trigger and capture trigger.

Figure 6-26: Example of basic timing operating as an input pulse interval measurement (MDmn0=0)



Note 1: m: unit number (m= 0) n: channel number (n=0~3)

Note 2: TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TI mn: TI mn pin input signal

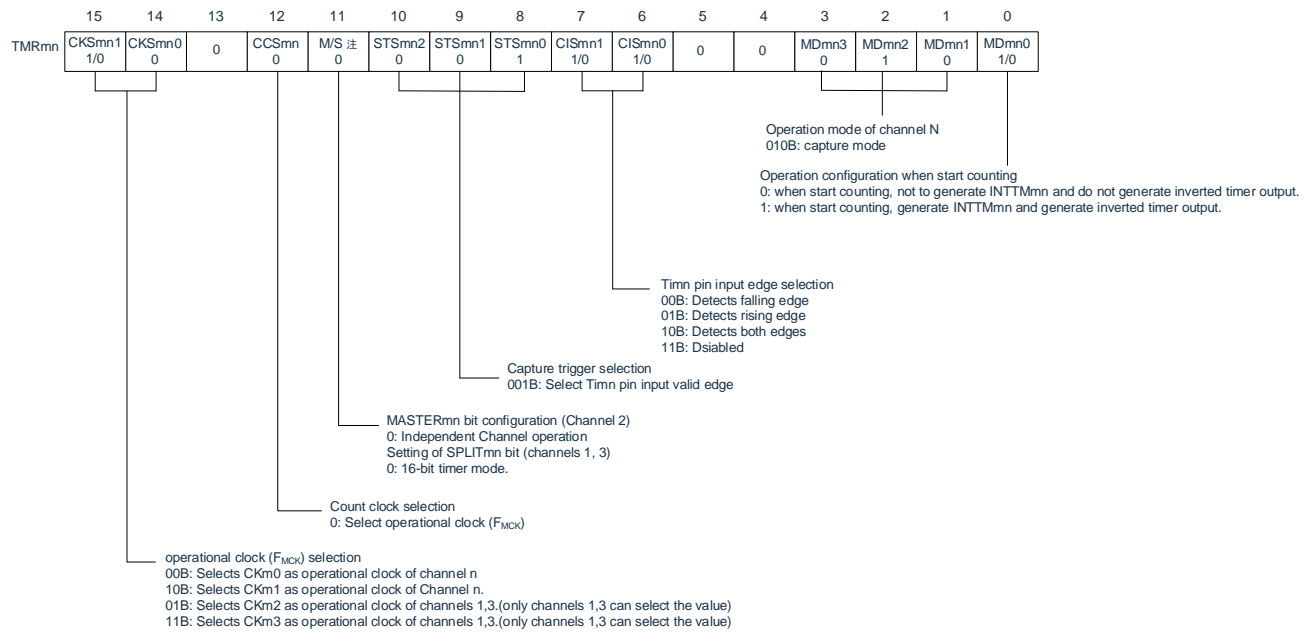
Note 3: TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit0 of timer status register mn (TSRmn)

Figure 6-27: Example of register contents setting in measuring input pulse interval

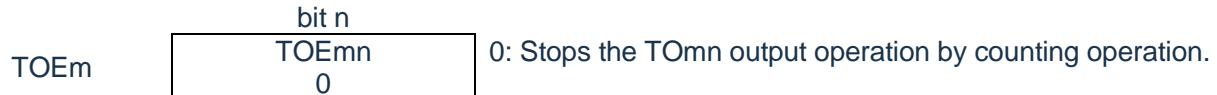
(a) Timer mode register mn (TMRmn)



(b) Timer output enable register m (TOEm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmn bit

Note 2: TMRm1, TMRm3: SPLITmn bit

Note 3: TMRm0: Fixed to "0".

Note 4: m: unit number (m=0) n: channel number (n=0~3)

Table 6-31: Procedure for input pulse interval measurement function

| | Software operation | Hardware status |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). | The channel is in the stop state. (Provides clock, and consumes some power) |
| Start operation | Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0". | The TEMn bit becomes "1" and starts counting. Clear the timer count register mn (TCRmn) to "0000H". When the MDmn0 bit of TMRmn register is "1", INTTMmn is generated. |
| In operation | The setting values of the CISmn1 bit and the CISmn0 bit of the TMRmn register can be changed. The TDRmn register can be read at any time. The TCRmn register can be read at any time. The TSRmn register can be read at any time. The setting of the TOMmn bit, the TOLmn bit, the TOMn bit and the TOEmn bit cannot be changed. | The counter (TCRmn) starts incremental counting from "0000H" and transfers (captures) the count value to the timer data register mn (TDRmn) if it detects an active edge on the TImn pin input or sets TSmn bit to "1". If the counter overflows, set the OVF bit of Timer Status Register mn (TSRmn). If the counter does not overflow, the OVF bit is cleared. Thereafter, repeat this operation. |
| Stop operation | Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit. | The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. The OVF bit of the TSRmn register remains unchanged. |
| Timer4 stop | Set the TM4mEN bit of the PER0 register to "0". | The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. |

Restart operation

Note: m: unit number (m= 0) n: channel number (n=0~3)

6.7.5 Operation as input signal high-/low-level width measurement

Note: When used as a LIN-bus support function, bit1 (ISC1) of the Input Switching Control Register (ISC) must be set to “1” and RxD0 should be used instead of TImn in the following description.

The signal width (high-/low-level width) of TImn can be measured by starting counting at one edge of the input to the TImn pin and capturing the count value at the other edge. The TImn signal width of the TImn output can be calculated using the following equation:

$$\text{Signal width of TImn input} = \text{period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{TDRmn captured value} + 1))$$

Note: Because the TImn pin inputs are sampled by the operation clock selected by the CKSmn bit of the Timer Mode Registermn (TMRmn), an error of 1 operation clock is generated.

In the Capture & Single Count mode, the timer count register mn (TCRmn) is used as an increment counter. If the channel start trigger bit (TSmn) of the timer channel start register m(TSm) is set to “1”, the TEMn bit becomes “1”, and the start edge detection wait state of the TImn pin is entered.

If the start edge of the TImn pin input (rising edge of the TImn pin input at the time of high-level width measurement) is detected, it is synchronized with the count clock and counts incrementally from “0000H”. Then, if an active capture edge is detected (falling edge of TImn pin input at the time of high-level width measurement), the count value is transferred to the Timer Data Register mn (TDRmn) and INTTMmn is output at the same time. If the counter overflows, the OVF bit of the Timer Status Register mn (TSRmn) is set to “1”. If the counter does not overflow, the OVF bit is cleared. The value of the TCRmn register changes to “Value passed to TDRmn register + 1”, and the start edge detection wait state of the TImn pin is entered. After that, continue the same operation.

While capturing the count value to the TDRmn register, the OVF bit of the TSRmn register is updated according to whether or not overflow occurs during the measurement, and the overflow status of the captured value can be confirmed.

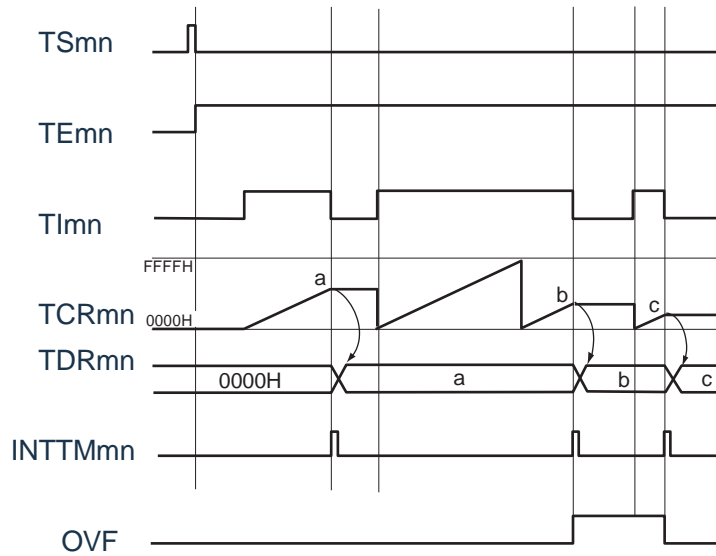
Even if the counter counts 2 or more complete cycles, the overflow is considered to have occurred and the OVF bit of the TSRmn register is set to “1”. However, when two or more overflows occur, the interval value cannot be measured normally by the OVF bit.

The CISmn1 and CISmn0 bits of the TMRmn register can be used to set whether the high-level width or low-level width of the TImn pin is to be measured. This function is designed to measure the input signal width of the TImn pin, so the TSmn bit cannot be set to “1” during the period when the TEMn bit is “1”.

CISmn1, CISmn0=10B of the TMRmn register: Measures the low-level width.

CISmn1, CISmn0=11B of the TMRmn register: Measures the high-level width.

Figure 6-28: Example of basic timing operating as high-/low-level width measurement of input signal



Note 1: m: unit number (m= 0) n: channel number (n=0~3)

Note 2: TS_{mn}: Bit n of timer channel start register m (TS_m)

TE_{mn}: Bit n of timer channel enable status register m (TE_m)

TI_{mn}: TI_{mn} pin input signal

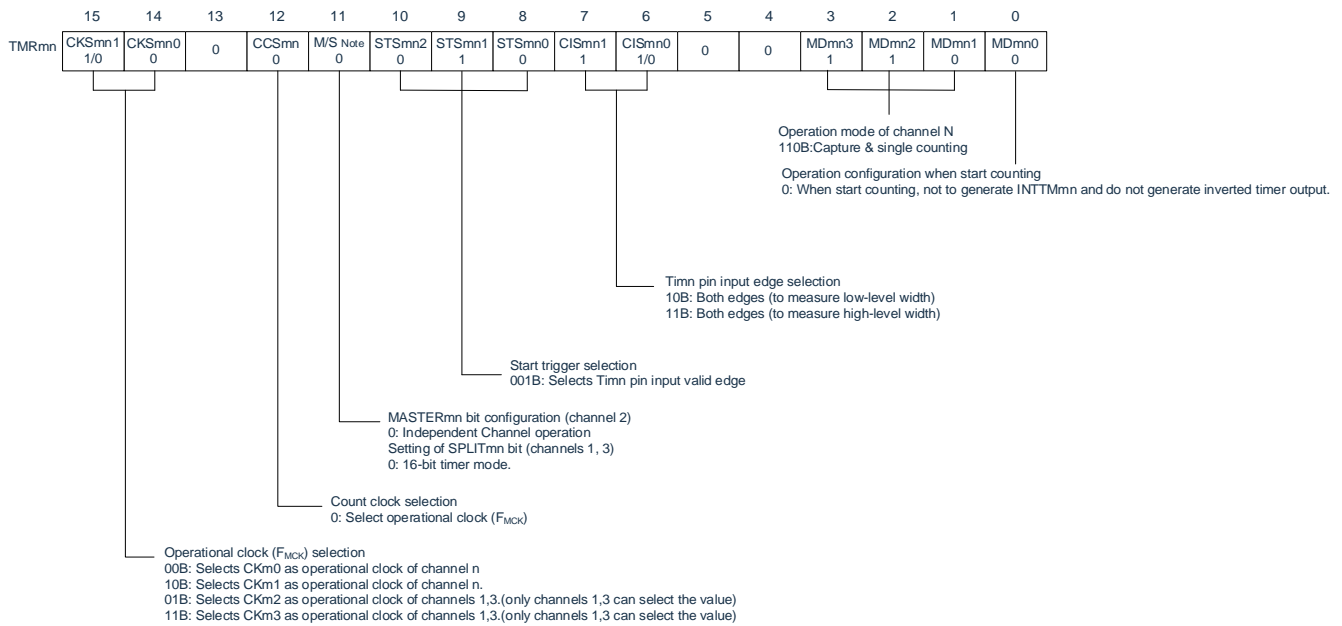
TCR_{mn}: Timer count register mn (TCR_{mn})

TDR_{mn}: Timer data register mn (TDR_{mn})

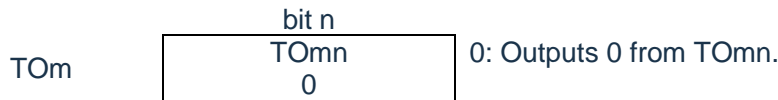
OVF: Bit 0 of timer status register mn (TSR_{mn})

Figure 6-29: Example of register contents setting in measuring high-/low-level width of input signal

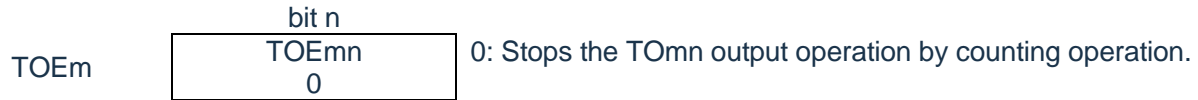
(a) Timer mode register mn (TMRmn)



(b) Timer output enable register m (TOEm)



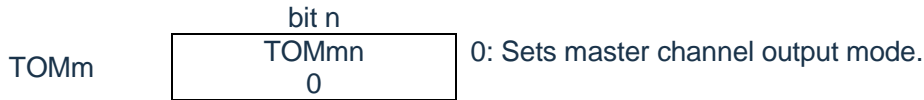
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmn bit

Note 2: TMRm1, TMRm3: SPLITmn bit

Note 3: TMRm0: Fixed to "0".

Note 4: m: unit number (m= 0) n: channel number (n=0~3)

Table 6-32: Procedure for high-/low-level width measurement function of input signal

| | Software operation | Hardware status |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | |
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). | The channel is in the stop state. (Provides clock, and consumes some power) |
| Start Operation | Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0". | The TEMn bit changes to "1" and enters the detection wait state for start triggering (detecting the active edge of the Tlmn pin input or setting the TSmn bit to "1"). |
| | Detect Tlmn pin input counting start edge. | Clear timer count register mn (TCRmn) to "0000H" and start incremental counting. |
| In operation | The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used. The setting of the TMRmn register, the TOMmn bit, the TOLmn bit, the Tomn bit and the TOEmn bit cannot be changed. | After the start edge of the Tlmn pin is detected, the counter (TCRmn) starts counting incrementally from "0000H". If the capture edge of the Tlmn pin is detected, the count value is transferred to the timer data register mn (TDRmn), and INTTMmn is generated. If the counter overflows, set the OVF bit of Timer Status Register mn (TSRmn). If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops counting before the start edge of the next Tlmn pin is detected. Thereafter, repeat this operation. |
| Stop operation | Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit. | The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. The OVF bit of the TSRmn register remains unchanged. |
| Timer4 stop | Set the TM4mEN bit of the PER0 register to "0". | The input clock of timer unit m is in the stop providing state. Initialize all circuits and the SFR for each channel. |

Restart operation

Note: m: unit number (m= 0) n: channel number (n=0~3)

6.7.6 Operation as delay counter

The count can be decremented by the active edge detection (external event) of the TImn pin input and INTTMMmn (timer interrupt) is generated at any set interval.

During the period when the TEMn bit is “1”, the TSmn bit can be set to “1” by software to start decreasing counting and generate INTTMMmn (timer interrupt) at any set interval.

The interrupt generation period can be calculated using the following equation:

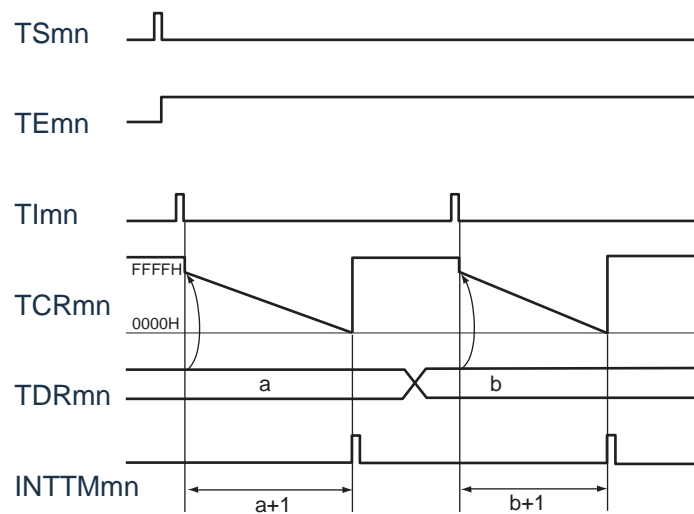
$$\text{INTTMMmn (timer interrupt) generation period} = \text{counting clock period} \times (\text{TDRmn set value} + 1)$$

In the single count mode, the timer count register mn (TCRmn) is used as a decrement counter.

If the channel start trigger bit (TSmn, TSHm1, TSHm3) of the timer channel start register m(TSm) is set to “1”, the TEMn bit, TEHm1 bit, TEHm3 bit become “1”, and the active edge detection wait state of the TImn pin is entered. An active edge detection via the TImn pin input starts the TCRmn register and loads the value of the Timer Data Register mn (TDRmn). The TCRmn register counts decreasingly from the value of the loaded TDRmn register by counting the clock. If TCRmn becomes “0000H”, INTTMMmn is output and counting is stopped until the next active edge of the TImn pin input is detected.

The TDRmn register can be rewritten at any time, and the rewritten TDRmn register value is valid from the next cycle.

Figure 6-30: Example of basic timing operating as a delay counter



Note 1: m: unit number (m= 0) n: channel number (n=0~3)

Note 2: TSmn: Bit n of timer channel start register m (TSm)

TEMn: Bit n of timer channel enable status register m (TEm)

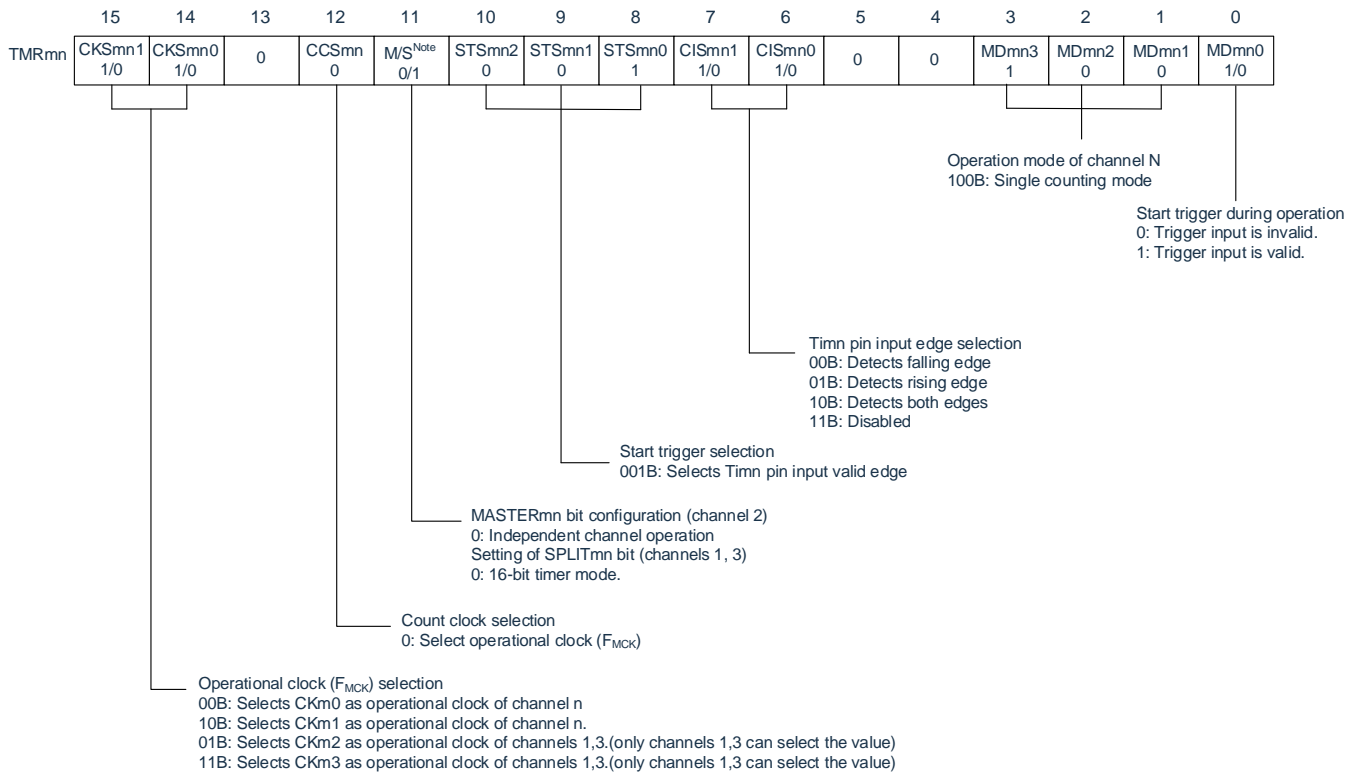
TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 6-31: Example of register contents setting for delay counter function

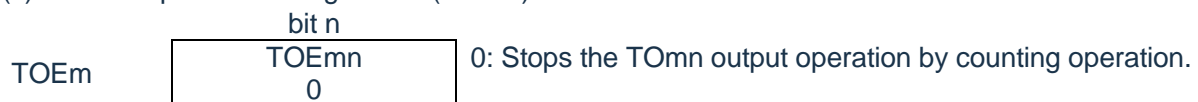
(a) Timer mode register mn (TMRmn)



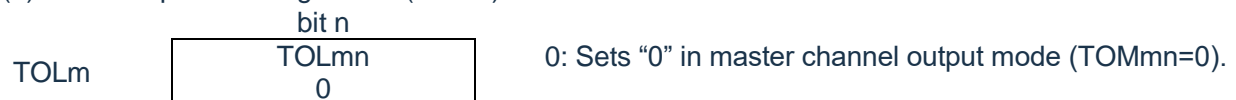
(b) Timer output enable register m (TOEm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmn bit

Note 2: TMRm1, TMRm3: SPLITmn bit

Note 3: TMRm0: Fixed to "0".

Note 4: m: unit number (m= 0) n: channel number (n=0~3)

Table 6-33: Procedure for delay counter function

| | Software operation | Hardware status |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| Initial setting of channels | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "0" (OFF) or "1" (ON). Set the timer mode register mn (TMRmn)(to determines the operating mode of the channel). Set the output delay time for the timer data register mn (TDRmn). Set the TOEmn bit to "0" and stop TOMn operation. | The channel is in a running stop state. (Provides clock, consumes some power) |
| Start Operation | Set the TSmn bit to "1". Since the TSmn bit is a trigger bit, it automatically returns to "0". | The TEMn bit turns into '1' and enter into start trigger (detect Timn pin input active edge or set TSmn bit to '1') detection waiting state. |
| | Start decreasing the count by detecting the next start trigger. <ul style="list-style-type: none"> The active edge of the TImn pin input. Set the TSmn bit to "1" by software. | Load the value of the TDRmn register into the Timer Count Register mn (TCRmn). |
| In operation | The setting of the TDRmn register can be changed at will. The TCRmn register can be read at any time. The TSRmn register is not used. | The counter (TCRmn) performs decremental counting. If TCRmn counts to "0000H", INTTMmn is generated and TCRmn is "1" until the next start trigger is detected (detecting an active edge on the TImn pin input or setting TSmn to "1"). The count is stopped when "0000H" is detected. |
| Stop operation | Set the TTmn bit to "1". The operation automatically returns to "0" because the TTmn bit is a trigger bit. | The TEMn bit becomes "0" and stops counting. The TCRmn register holds the count value and stops counting. |
| Timer4 stop | Set the TM4mEN bit of the PER0 register to "0". | The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. |

Restart operation

Note: m: unit number (m= 0) n: channel number (n=0~3)

6.8 Multi-channel linkage operation function for general purpose timer unit

6.8.1 Operation as single trigger pulse output function

Using the 2 channels in pairs, a single trigger pulse with any delay pulse width can be generated from the input of the TImn pin. The delay and pulse width can be calculated using the following equation:

$$\begin{aligned}\text{Delay} &= \{\text{TDRmn (master) set value} + 2\} \times \text{counting clock period} \\ \text{Pulse width} &= \{\text{TDRmp (slave) set value}\} \times \text{counting clock period}\end{aligned}$$

In single count mode, the master channel operates and counts the delay. By detecting a start trigger, the timer count register mn (TCRmn) of the master channel starts to operate and loads the value of timer data register mn (TDRmn). The TCRmn register counts decreasingly from the value of the loaded TDRmn register by counting the clock. If TCRmn becomes “0000H”, INTTMmn is output and counting stops before the next start trigger is detected.

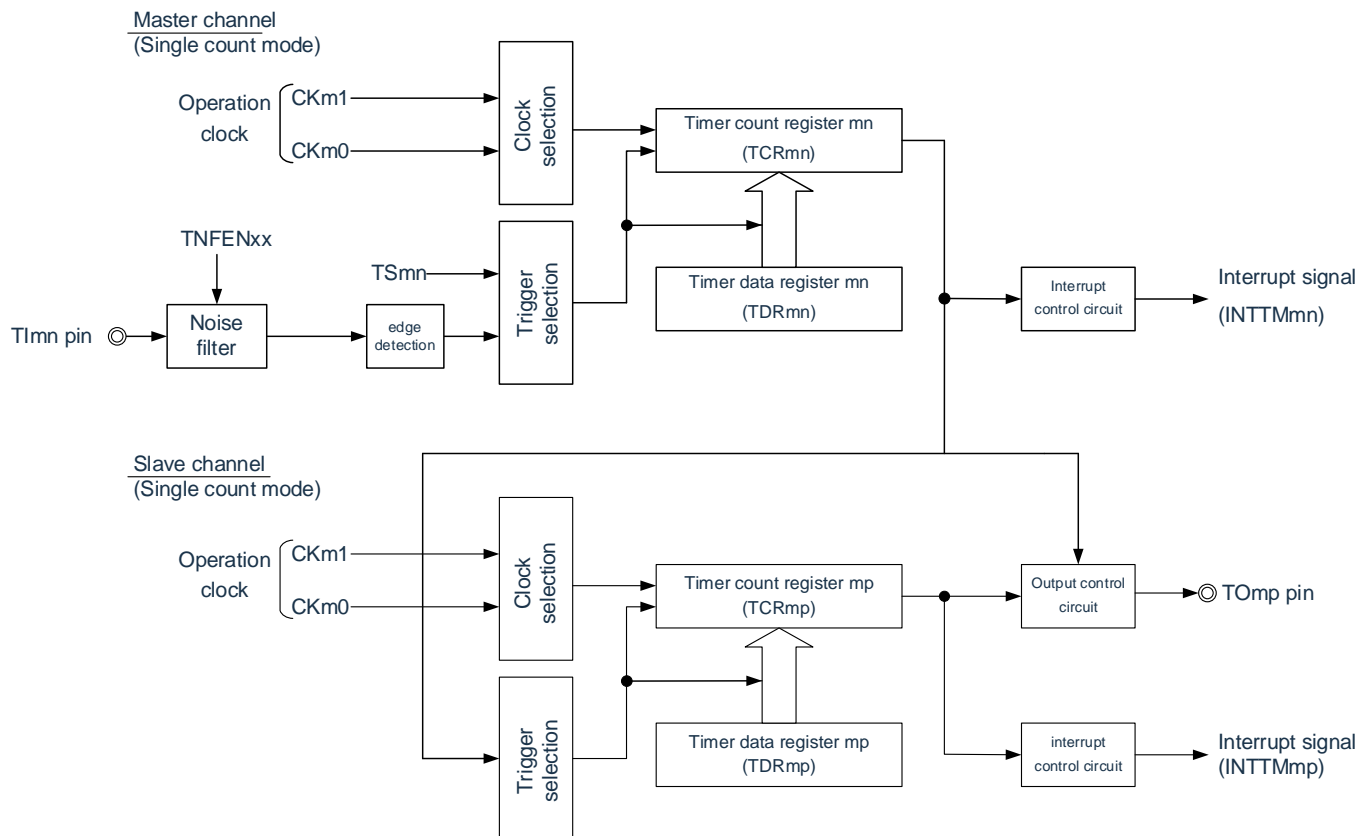
In single count mode, the slave channel operates and counts the pulse width. The INTTMmn of the master channel is used as the start trigger and the TCRmp register of the slave channel is started and loaded with the value of the TDRmp register. The TCRmp register counts decreasingly from the value of the loaded TDRmp register by counting the clock. If the count value becomes “0000H”, INTTMmp is output and counting is stopped until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes valid after INTTMmn has been generated from the master channel and after 1 count clock, if TCRmp becomes “0000H”, it becomes invalid.

The software operation (TSmn=1) can also be used as a start trigger to output a single trigger pulse without using the TImn pin input.

Note 1: Because the TDRmn register of the master channel and the TDRmp register of the slave channel have different loading timings, if the TDRmn register and the TDRmp register are rewritten during counting, they may compete with the loading timings and output an abnormal waveform. The TDRmn register must be rewritten after generating INTTMmn and the TDRmp register must be rewritten after generating INTTMmp.

Note 2: m: unit number (m=0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

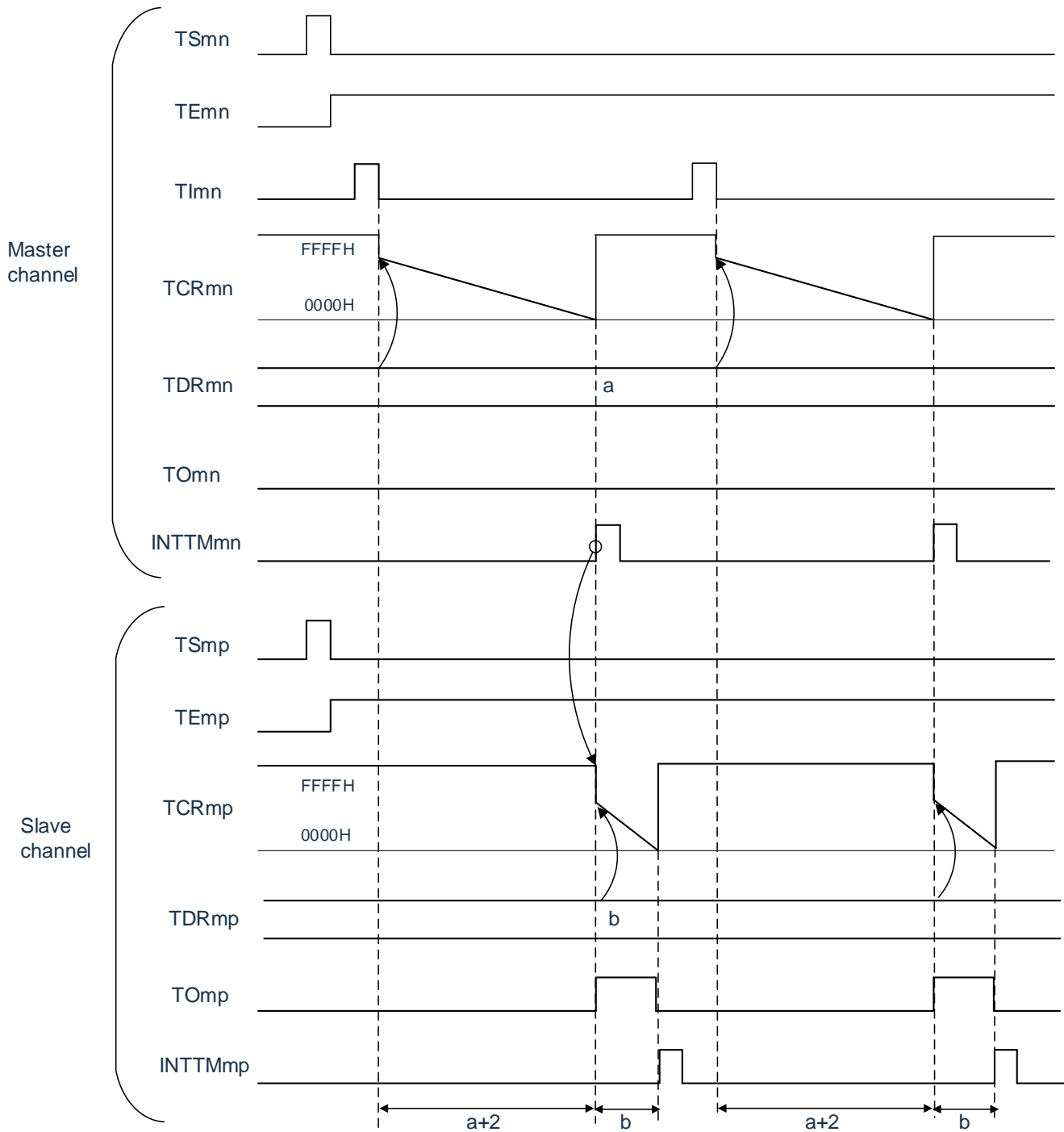
Figure 6-32: Block diagram of operation as single trigger pulse output function



Note: m: unit number (m= 0) n: master channel number (n=0, 2)

p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 6-33: Example of basic timing operating as a single trigger pulse output function

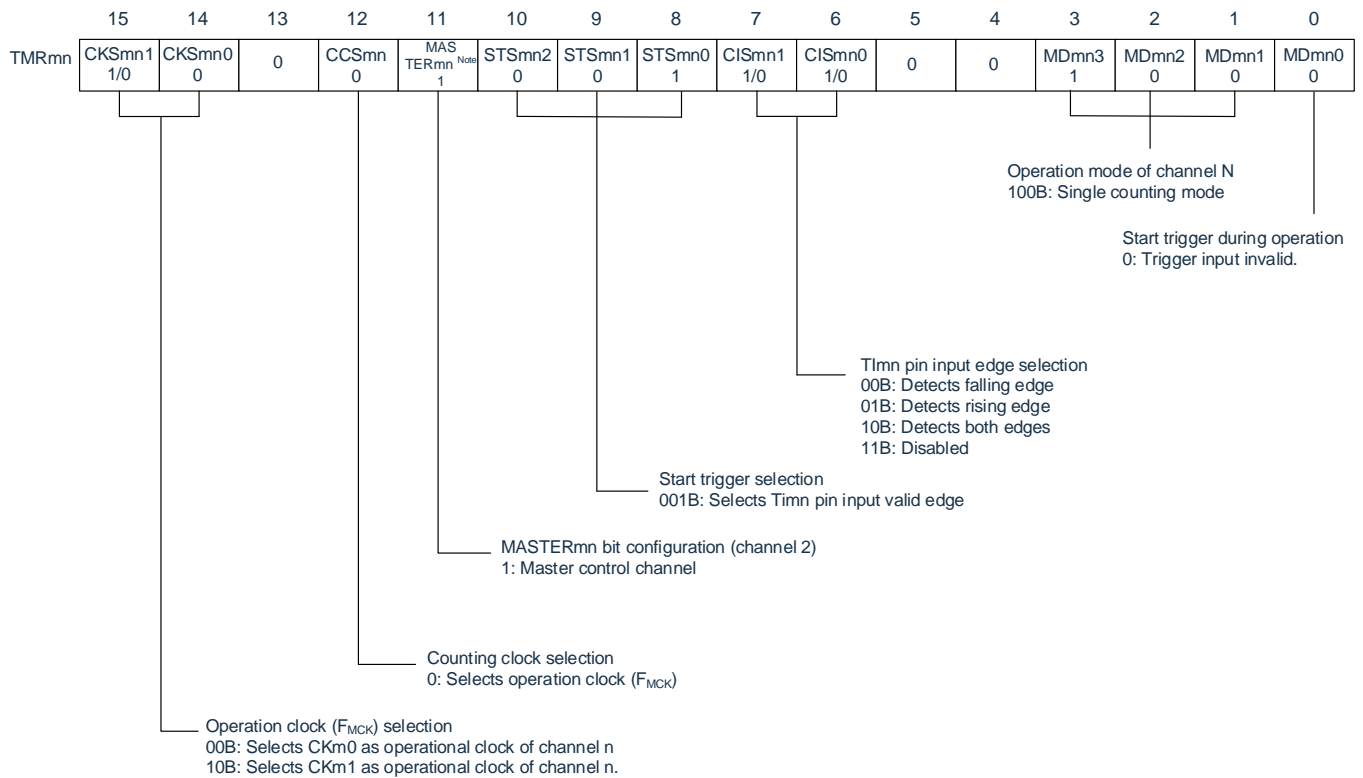


Note 1: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Note 2: TSmn, TSmp: Bit n of timer channel start register m (TSm), p
 TEmn, TEmp: Bit n of timer channel enable status register m (TEm), p
 TImn, TImp: Input signals of TImn pin and TImp pin
 TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
 TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
 TOmn, TOmp: Output signals of TOmn pin and TOmp pin

Figure 6-34: Example of register contents setting for single trigger pulse output function (master channel)

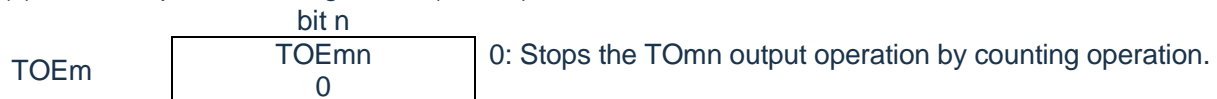
(a) Timer mode register mn (TMRmn)



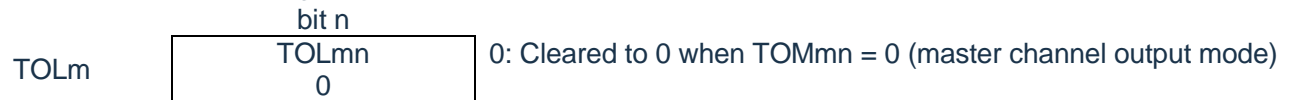
(b) Timer output enable register m (TOEm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



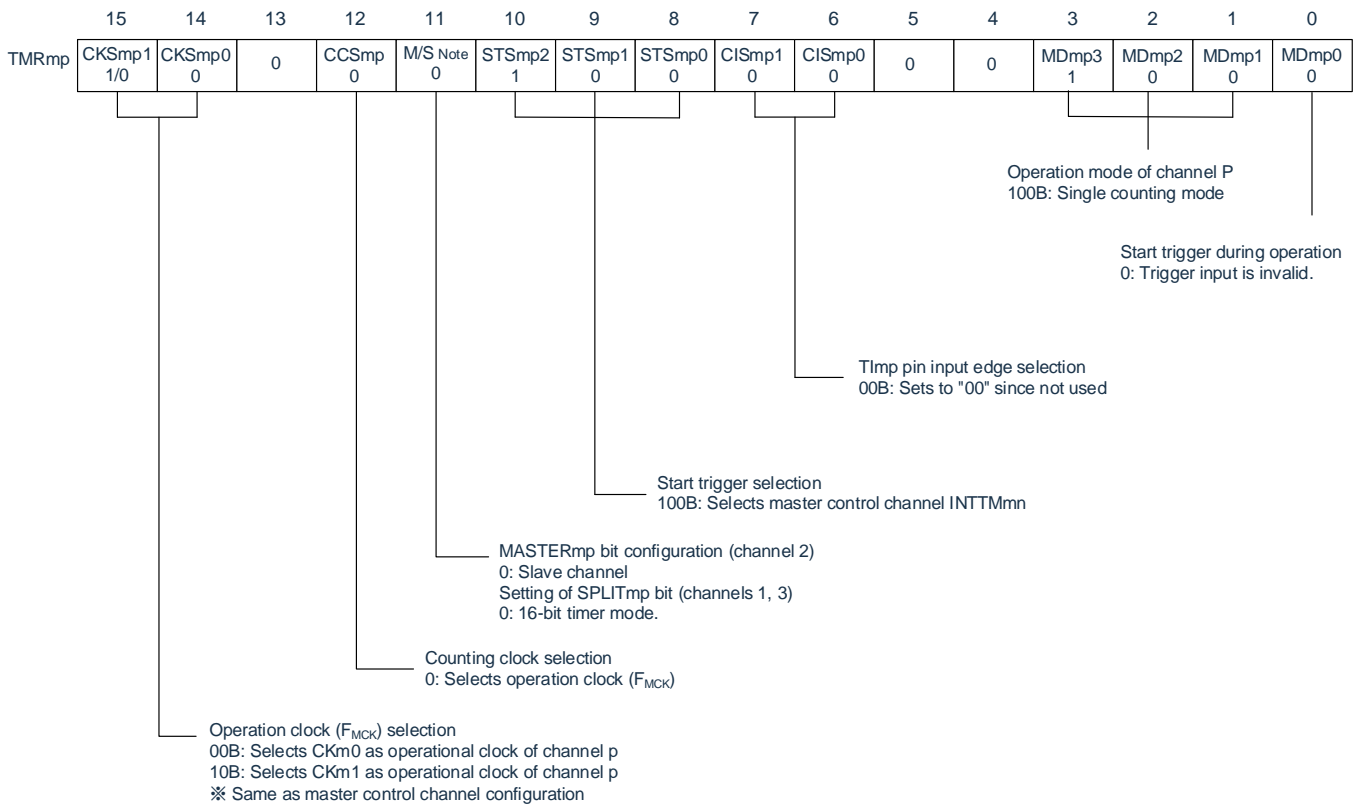
Note 1: m: unit number (m= 0, 1) n: master channel number (n=0, 2)

Note 2: TMRm2: MASTERmn=1

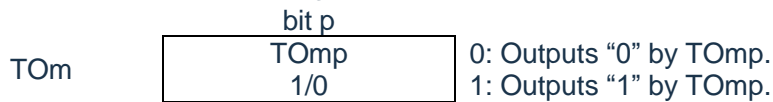
TMRm0: Fixed to "0".

Figure 6-35: Example of register contents setting for single trigger pulse output function (slave channel)

(a) Timer mode register mp (TMRmp)



(b) Timer output enable register m (TOEm)



(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmp bit

TMRm1, TMRm3: SPLITmp bit

Note 2: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Table 6-34: Procedure for single trigger pulse output function(1/2)

| | Software operation | Hardware status |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the corresponding bit of the Noise Filter Enable Register (NFEN1) to "1". Set the timer mode registers mn and mp (TMRmn, TMRmp) for the 2 channels used (to determine the operation mode of the channel). Set the output delay time for the timer data register mn (TDRmn) of the master channel, and set the pulse width for the TDRmp register of the slave channel. | The channel is in the stop state. (Provides clock, and consumes some power) |
| | Slave channel setting Set TOMmp bit of the timer output mode register m (TOMm) to "1" (slave channel output mode). Set the TOLmp bit. Set the TOmp bit to determine the initial level of the TOmp output. Set the TOEmp bit to "1" and enable TOmp output. Set the Port Register and Port Mode Register to "0". | The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TOmp initial set level is output. The TOmp remains unchanged because the channel is in the stop state. The TOmp pin outputs the level set by the TOmp. |

Table 6-34: Procedure for single trigger pulse output function(2/2)

| | Software operation | Hardware status |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Start operation | Set the TOEmp bit (slave) to "1" (restart operation only). Set the TSmn (master) and TSmp (slave) bits of the Timer Channel Start Register m (TSm) to "1" at the same time. Since the TSmn bit and the TSmp bit are trigger bits, they automatically return to "0". | The TEMn and TEm bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating. |
| | Count operation of the master channel is started by start trigger detection of the master channel • Detects the TImn pin input valid edge • Sets the TSmn bit of the master channel to 1 by software Note. | Master channel starts counting. |
| In operation | Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOM and TOEm registers by slave channel can be changed. | Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and stops counting until the next TImn pin input. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated. |
| Stop operation | The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits. | TEMn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status. |
| | The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. | The TOmp pin outputs the TOmp set level. |
| Timer4 stop | To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary: Setting not required. | The TOmp pin output level is held by port function. |
| | The TM4mEN bit of the PER0 register is cleared to 0. | The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR of each channel. |

Note: The TSmn bit of the slave channel cannot be set to "1".

Note: m: unit number (m= 0) n: master channel number (n=0) p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than n)

6.8.2 Operation as PWM function

By using the 2 channels in pairs, pulses of any period and duty cycle can be generated. The period and duty cycle of the output pulses can be calculated using the following equations:

Pulse period = {TDRmn (master) set value +1} × counting clock period
Duty cycle [%] = {TDRmp (slave) set value} / {TDRmn (master) set value +1} ×100
0% output: TDRmp (slave) set value = 0000H
100% output: TDRmp (slave) set value ≥ {TDRmn (master) set value +1}
Output: TDRmp (slave) set value ≥ {TDRmn (master) set value +1}

Note: When the set value of TDRmp (slave) > {Set value of TDRmn (master) +1}, the duty cycle exceeds 100% but is 100% output.

The master channel is used as the interval timer mode. If the channel start trigger bit (TSMn) of the timer channel start register m (TSM) is set to “1”, an interrupt (INTTMmn) is output, and then the set value of the timer data register mn (TDRmn) is loaded into the timer count register mn (TCRmn), and the count is decremented by the count clock. When the count reaches “0000H”, the value of the TDRmn register is loaded into the TCRmn register again after the INTTMmn is output, and the count is decremented. Thereafter, this operation is repeated before setting the channel stop trigger bit (TTmn) of the timer channel stop register m (TTM) to “1”.

When used as PWM function, the master channel decrements the count and the period until “0000H” is counted as the PWM output (TOmp) period. The slave channel is used in single count mode. The value of TDRmp register is loaded into TCRmp register with INTTMmn of the master channel as the start trigger, and the count is decremented until “0000H”. When the count reaches “0000H”, INTTMmp is output and the next start trigger (INTTMmn of the master channel) is waited.

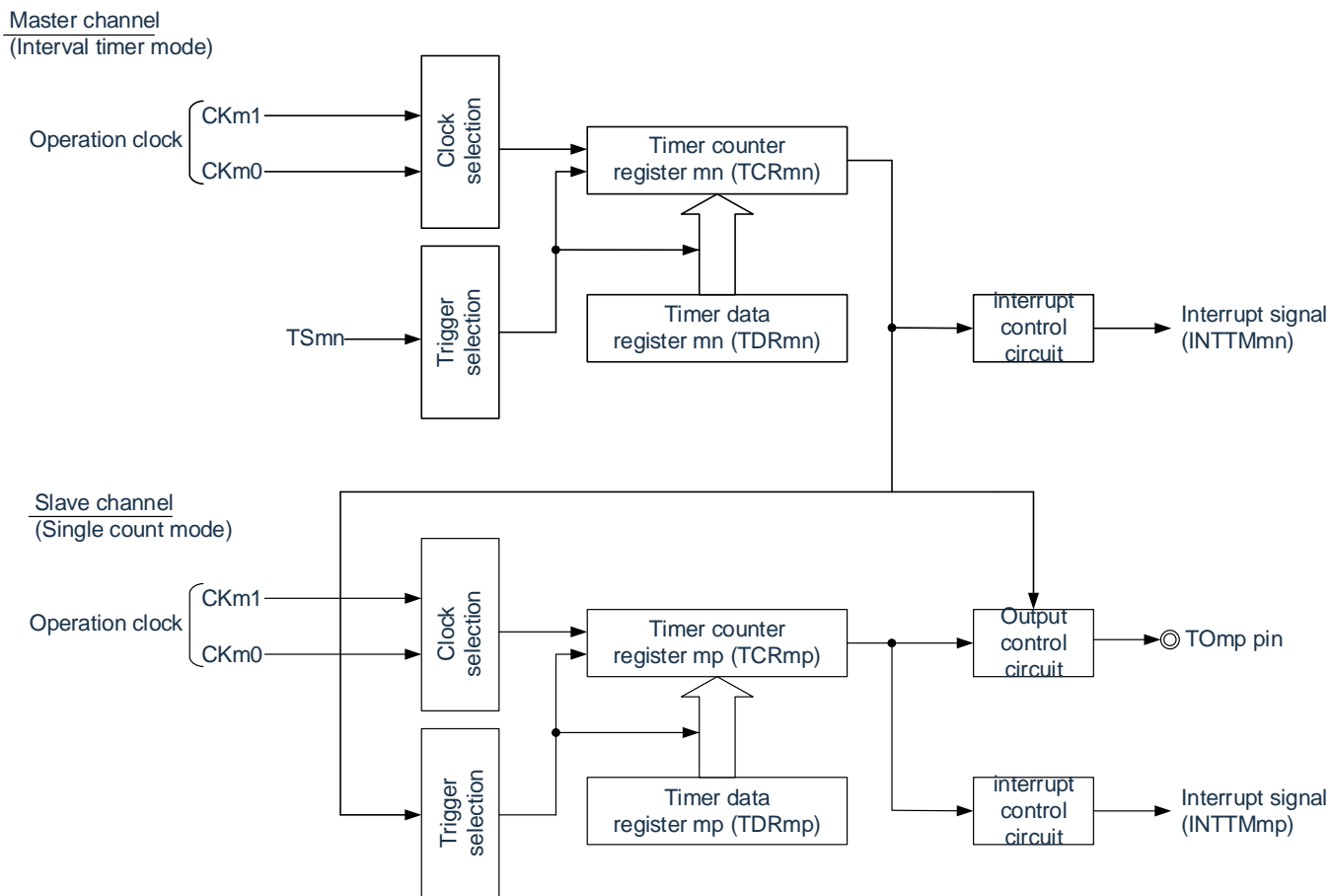
When used as PWM function, the slave channel decrements the count and the duty cycle of the PWM output (TOmp) for the period until “0000H” is counted.

After INTTMmn is generated from the master channel and 1 clock has elapsed, the PWM output (TOmp) becomes active and it becomes invalid when the value of TCRmp register of the slave channel is “0000H”.

Note 1: To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

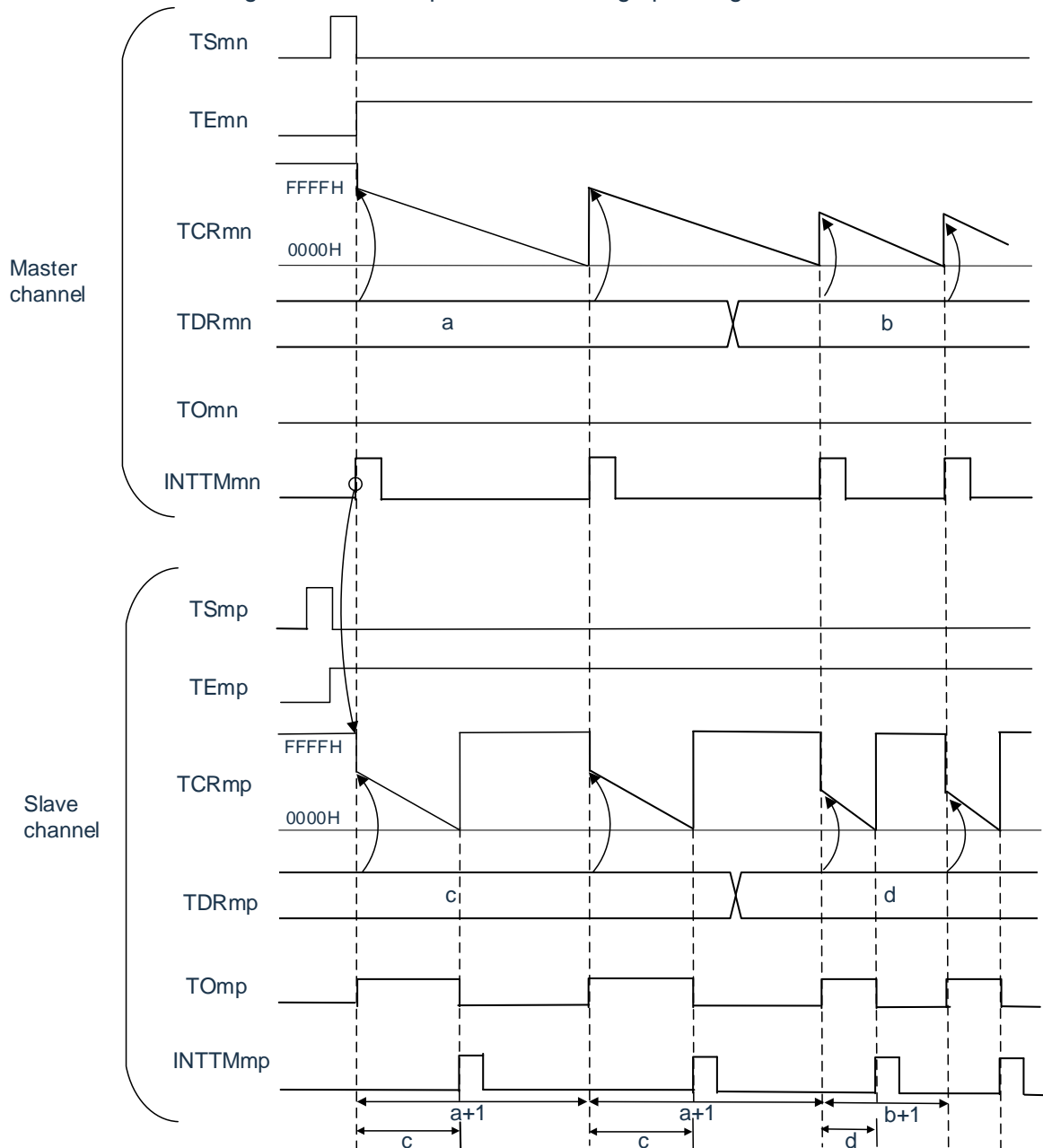
Note 2: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 6-36: Block diagram of operation as PWM function



Note: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Figure 6-37: Example of basic timing operating as PWM function



Note 1: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Note 2: TSmn, TSmp: Bit n of timer channel start register m (TSM), p

TEmn, TEmn: Bit n of timer channel enable status register m (TEM), p

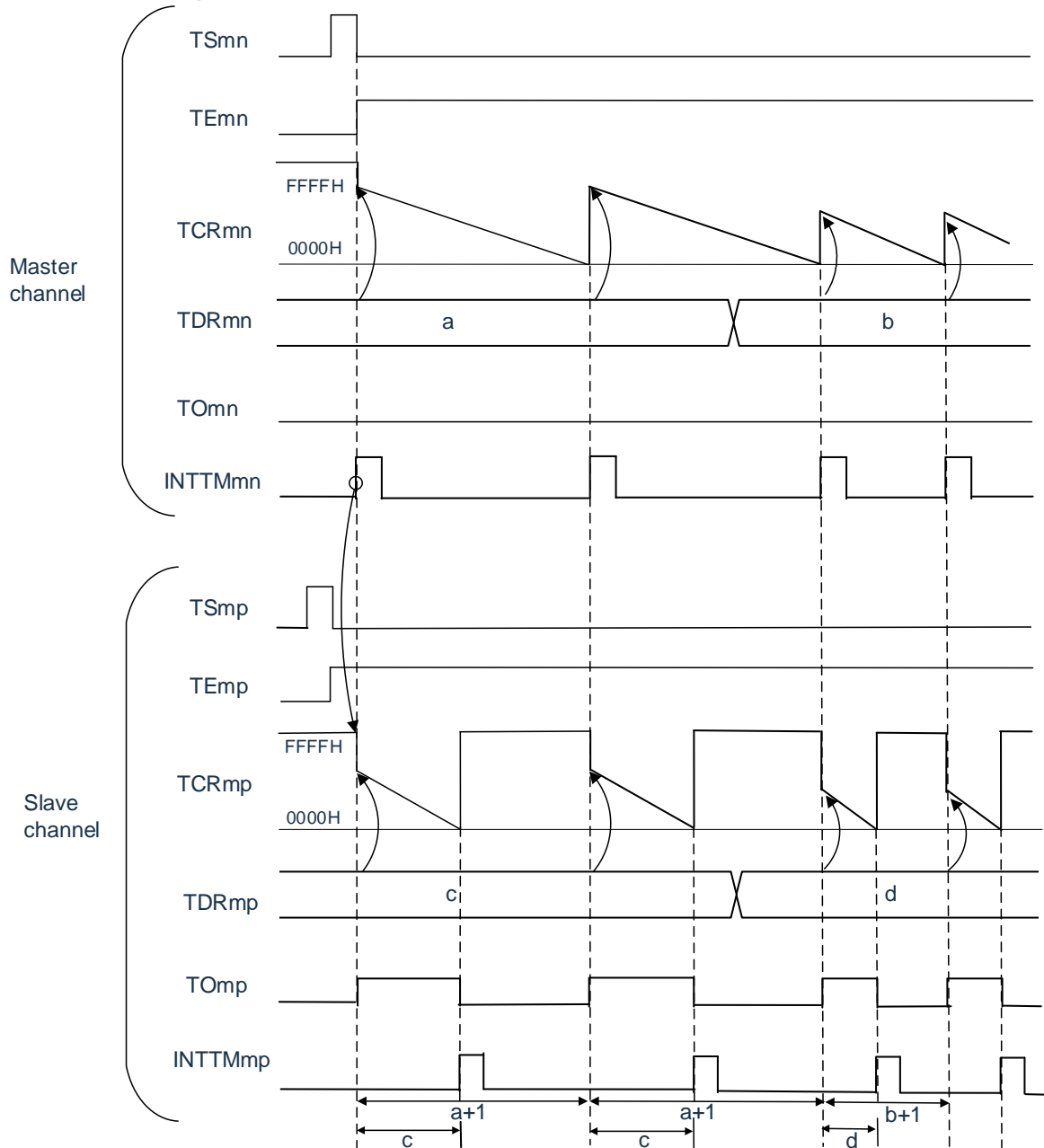
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

TOmn, TOmp: Output signals of TOmn pin and TOmp pin

Figure 6-38: Example of basic timing operating as PWM function

(a) Timer mode register mn (TMRmn)



(b) Timer output enable register m (TOEm)

bit n
TOMn
0
0: Outputs 0 from TOMn.

(c) Timer output enable register m (TOEm)

bit n
TOEmn
0
0: Stops the TOMn output operation by counting operation.

(d) Timer output level register m (TOLm)

bit n
TOLmn
0
0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

| | | | | |
|-------|-----------------------------------------------------------|-------|---|-------------------------------------|
| | bit n | | | |
| TOMm | <table><tr><td>TOMmn</td></tr><tr><td>0</td></tr></table> | TOMmn | 0 | 0: Sets master channel output mode. |
| TOMmn | | | | |
| 0 | | | | |

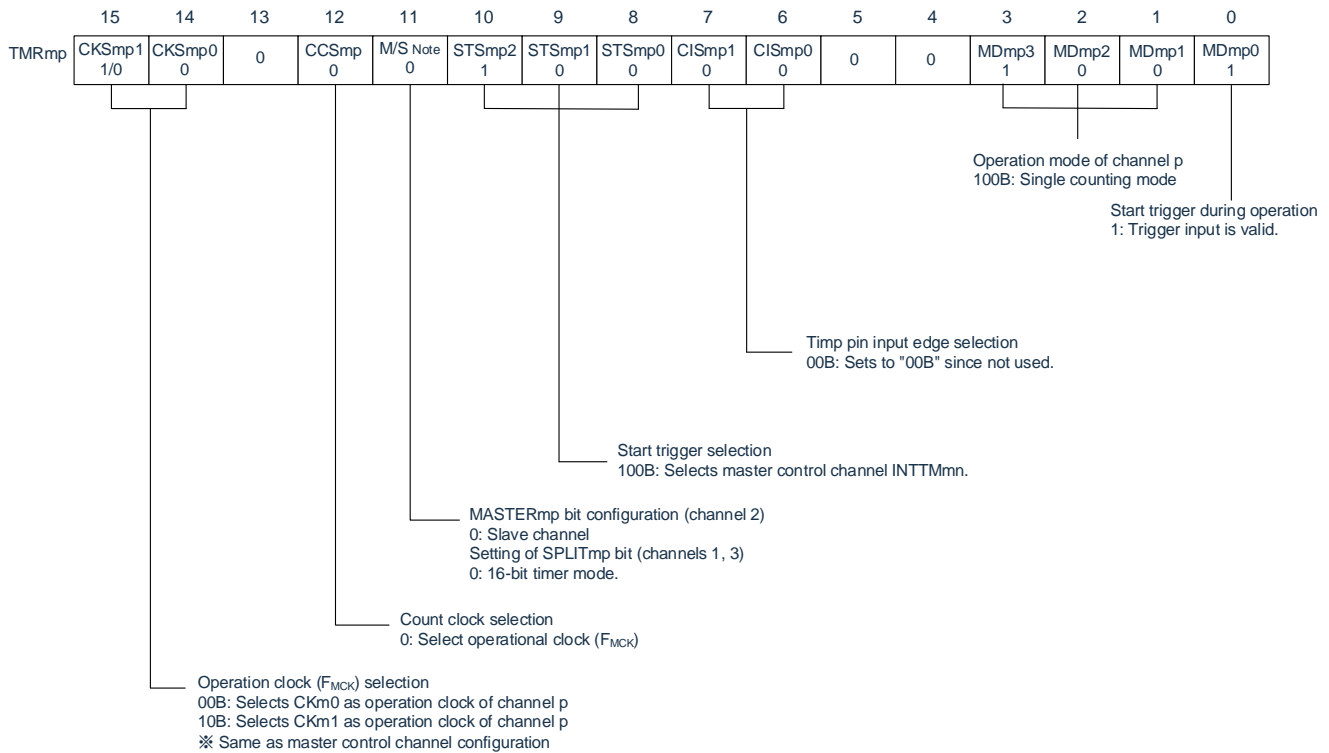
Note 1: m: unit number (m=0, 1) n: master channel number (n=0, 2)

Note 2: TMRm2: MASTERmn=1

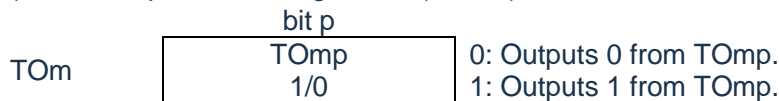
TMRm0: Fixed to "0".

Figure 6-39: Example of register contents setting for PWM function (slave channel)

(a) Timer mode register mp (TMRmp)



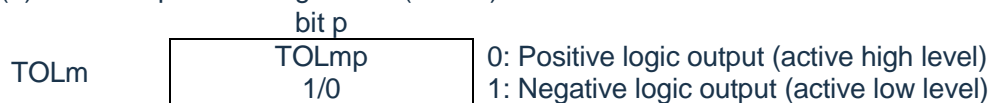
(b) Timer output enable register m (TOEm)



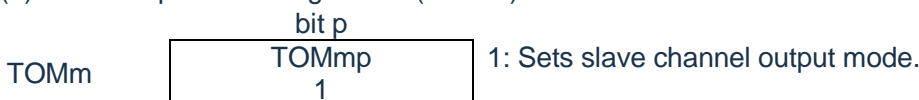
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note 1: TMRm2: MASTERmp bit

Note 2: TMRm1, TMRm3 : SPLITmp bit

Note 3: m: unit number (m= 0) n: master channel number (n=0, 2) p: slave channel number (n=0: p=1, 2, 3, n=2: p=3)

Table 6-35: Procedure for the PWM function(1/2)

| | Software operation | Hardware status |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 ~ CKm3. | |
| Initial setting of channels | Set the timer mode registers mn and mp (TMRmn, TMRmp) for the 2 channels used (to determine the operation mode of the channel). Set the interval (period) value for the timer data register mn (TDRmn) for the master channel and the duty cycle value for the TDRmp register for the slave channel. | The channel is in the stop state. (Provides clock, and consumes some power) |
| | Slave channel setting Set TOMmp bit of the timer output mode register m (TOMm) to "1" (slave channel output mode). Set the TOLmp bit. Set the TOmp bit to determine the initial level of the TOmp output. Set the TOEmp bit to "1" and enable TOmp output. Set the Port Register and Port Mode Register to "0". | The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the initially set level of TOmp is output. The TOmp remains unchanged because the channel is in the stop state. The TOmp pin outputs the level set by the TOmp. |

Table 6-35: Procedure for the PWM function(2/2)

| | Software operation | Hardware status |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Restart operation | Start operation Set the TOEmp bit to "1" (only limited to restart operation). Set both the TSmn bit (master) and TSmp bit (slave) of the timer channel start register m (TSm) to "1". The operation automatically returns to "0" because the TSmn and TSmp bits are trigger bits. | The TEMn and TEm bits become "1". The master channel starts counting and generates INTTMmn. With this as a trigger, the slave channel also starts counting. |
| | Run operation The setting values of the TMRmn and TMRmp registers and the TOMmn bit, TOMmp bit, TOLmn bit, and TOLmp bit cannot be changed. Able to change the setting value of the TDRmn register and the TDRmp register after the master channel has generated INTTMmn. The TCRmn and TCRmp registers can be read at any time. The TSRmn and TSRmp registers are not used. | The master channel loads the value of the TDRmn register into the timer count register mn (TCRmn) and perform decremental counting. If TCRmn counts till "0000H", then generating INTTMmn. At the same time, load the TDRmn register value into the TCRmn register and restart decremental counting. The slave channel use INTTMmn of master channel as a trigger, load the TDRmp register value into the TCRmp register and counter start decremental counting. After INTTMmn is output from the master channel and one count clock has elapsed, the output level of TOMP is set to an active level. Then, if TCRmp counts to "0000H", it stops counting after setting the output level of TOMP to an invalid level. Thereafter, repeat this operation |
| | Stop operation Set the TTmn bit (master) and TTmp bit (slave) to "1" at the same time. The operation automatically returns to "0" because the TTmn and TTmp bits are trigger bits. | TEMn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOMP output is not initialized but holds current |
| | Set the TOEmp bit of slave channel to "0" and set the value for the TOMP bit. | The TOMP pin outputs the TOMP set level. |
| | Timer4 stop To maintain the output level of the TOMP pin: Set TOMP bit to "0" after setting the value to be held for the port register. When holding the TOMP pin output level is not necessary: No need to set. Set the TM4mEN bit of the PER0 register to "0". | The TOMP pin output level is held by port function. The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TOMn bit becomes "0" and TOMP pin becomes port function) |

Note: m: unit number (m= 0) n: master channel number (n=0)

p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than n)

6.8.3 Operation as multiple PWM output function

This is a function that extends the PWM function and uses multiple slave channels for multiple PWM outputs with different duty cycles. For example, when using 2 slave channels in pairs, the period and duty cycle of the output pulse can be calculated by using the following equation:

$$\begin{aligned} \text{Pulse period} &= \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times \text{count clock period} \\ \text{Duty cycle1}[\%] &= \{\text{TDRmp}(\text{slave 1}) \text{ set value}\} / \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times 100 \\ \text{Duty cycle2}[\%] &= \{\text{TDRmq}(\text{slave 2}) \text{ set value}\} / \{\text{TDRmn}(\text{master}) \text{ set value} + 1\} \times 100 \end{aligned}$$

Note: When the set value of TDRmp (slave 1) > {the set value of TDRmn (master) + 1} or {the set value of TDRmq (slave 2)} > {the set value of TDRmn (master) + 1}, the duty cycle exceeds 100%, but is 100% output.

In interval timer mode, the timer count register mn (TCRmn) of the master channel operates and counts the period. In single count mode, the TCRmp register of slave channel 1 operates and counts the duty cycle and outputs the PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = "0000H", the TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes valid after INTTMmn has been generated from the master channel and after 1 count clock, if TCRmp becomes "0000H", it becomes invalid.

In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in single count mode, counts the duty cycle, and outputs a PWM waveform from the TOMq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = "0000H", the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of the TOMq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to 3 types of PWM signals can be output at the same time.

Note 1: To rewrite the timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1 at the same time, at least 2 write accesses are required. Because the values of TDRmn register and TDRmp register are loaded into the TCRmn register and TCRmp register when the master channel generates INTTMmn, the TOmp pin cannot output the expected waveform if rewriting is performed before and after the master channel generates INTTMmn respectively. Therefore, to rewrite both the master TDRmn register and the slave TDRmp register, these two registers must be rewritten immediately after the master channel generates INTTMmn (the same applies to the TDRmq register of slave channel 2).

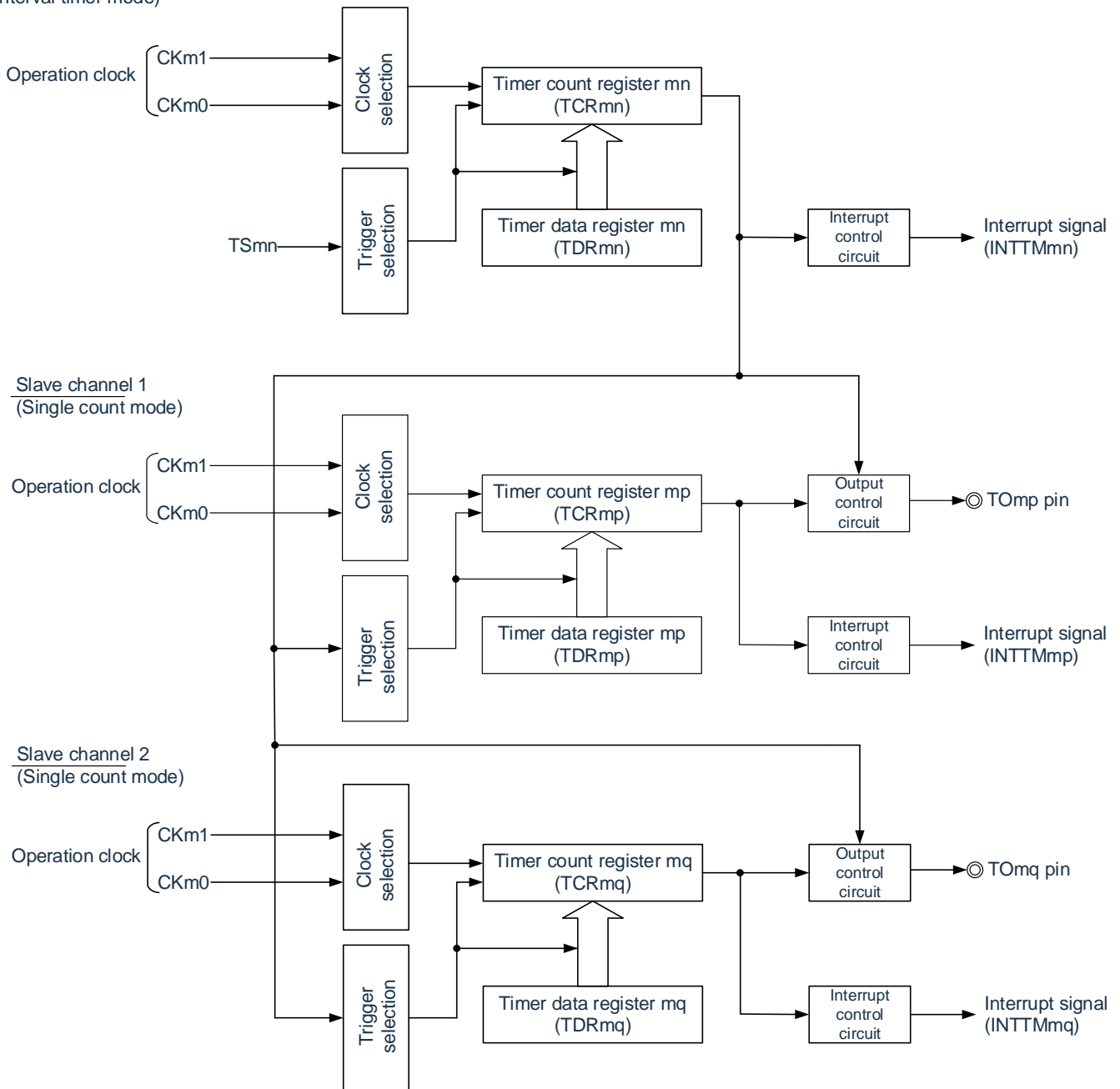
Note 2: m: unit number (m= 0) n: master channel number (n=0)

p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than n)

Figure 6-40: Block diagram of operation as multiple PWM output function (output two types of PWMs)

Master channel

(Interval timer mode)



Note: m: unit number (m= 0) n: master channel number (n=0)

p: Slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than

n)

The diagram illustrates the timing relationships for the SPI interface across three channels: Master channel, Slave channel 1, and Slave channel 2. The signals shown are:

- Master channel:** TSmn (Master chip select), TEmn (Master enable), TCRmn (Master transmit/receive data), TDRmn (Master transmit/receive data), TOMn (Master transmit/receive data), and INTTMmn (Master interrupt).
- Slave channel 1:** TSmp (Slave chip select), TEmrp (Slave enable), TCRmp (Slave transmit/receive data), TDRmp (Slave transmit/receive data), TOMp (Slave transmit/receive data), and INTTMmp (Slave interrupt).
- Slave channel 2:** TSmq (Slave chip select), TEmq (Slave enable), TCRmq (Slave transmit/receive data), TDRmq (Slave transmit/receive data), TOMq (Slave transmit/receive data), and INTTMmq (Slave interrupt).

The diagram shows the data flow and timing for the Master channel, Slave channel 1, and Slave channel 2. The Master channel signals are shown in the top section, Slave channel 1 in the middle, and Slave channel 2 in the bottom. The timing parameters are defined as follows:

- a:** Master channel data transfer time (from TEmn to TDRmn).
- b:** Master channel data transfer time (from TEmn to TCRmn).
- c:** Slave channel 1 data transfer time (from TEmrp to TDRmp).
- d:** Slave channel 1 data transfer time (from TEmrp to TCRmp).
- e:** Slave channel 2 data transfer time (from TEmrq to TDRmq).
- f:** Slave channel 2 data transfer time (from TEmrq to TCRmq).

The diagram also shows the timing for the interrupt signals (INTTMmn, INTTMmp, INTTMmq) and the enable signals (TEmn, TEmrp, TEmrq). The data transfer times (a, b, c, d, e, f) are shown as the time between the enable signal and the data transfer completion.

p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than

n)

Note 2: TSmn, TSmp, TSmq: Bit n of timer channel start register m (TSm), p, q

TEmn, TEmq, TEmq: Bit n of timer channel enable status register m (TEm), p, q

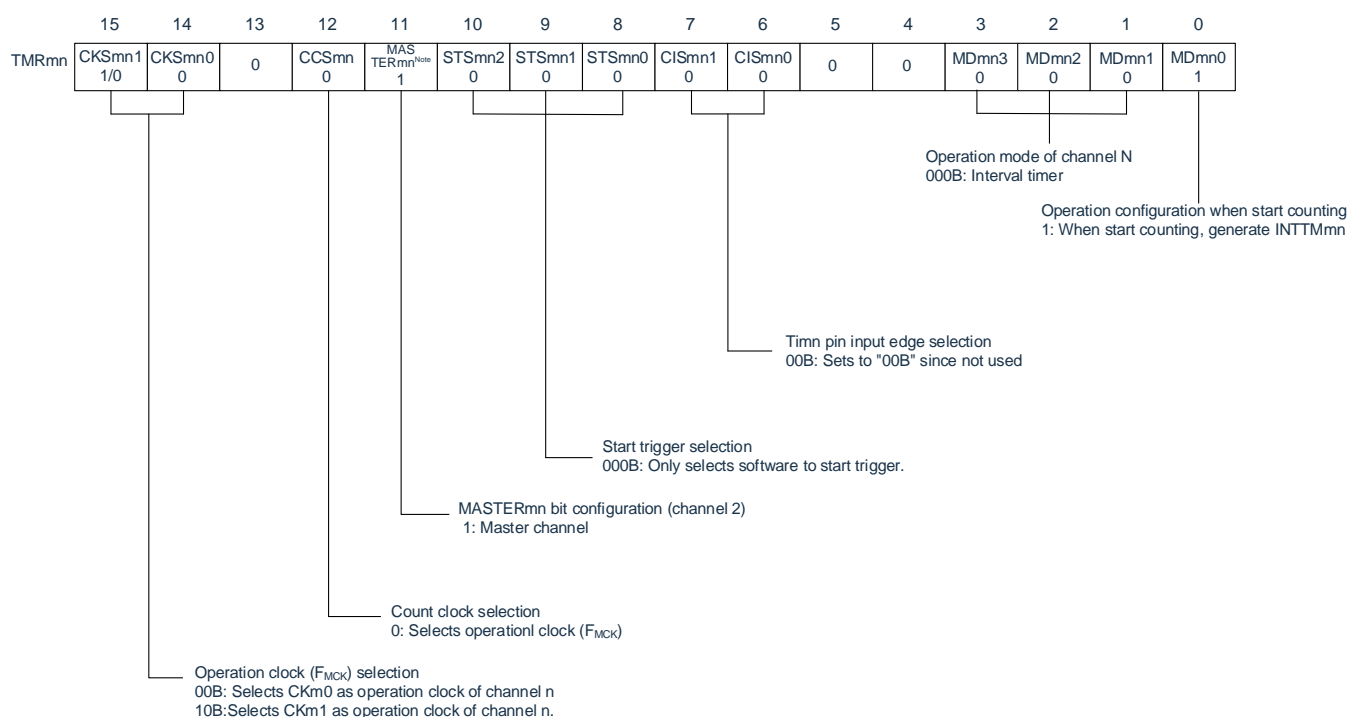
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)

TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)

TOmn, TOmp, TOmq: TOmn, TOmp, TOmq pin output signals

Figure 6-42: Example of register contents setting for multiple PWM output function (master channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output enable register m (TOEm)

bit n
TOm TOMn
0 0: Outputs 0 from TOMn.

(c) Timer output enable register m (TOEm)

bit n
TOEm TOEmn
0 0: Stops the TOMn output operation by counting operation.

(d) Timer output level register m (TOLm)

bit n
TOLm TOLmn
0 0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

bit n
TOMm TOMmn
0 0: Sets master channel output mode.

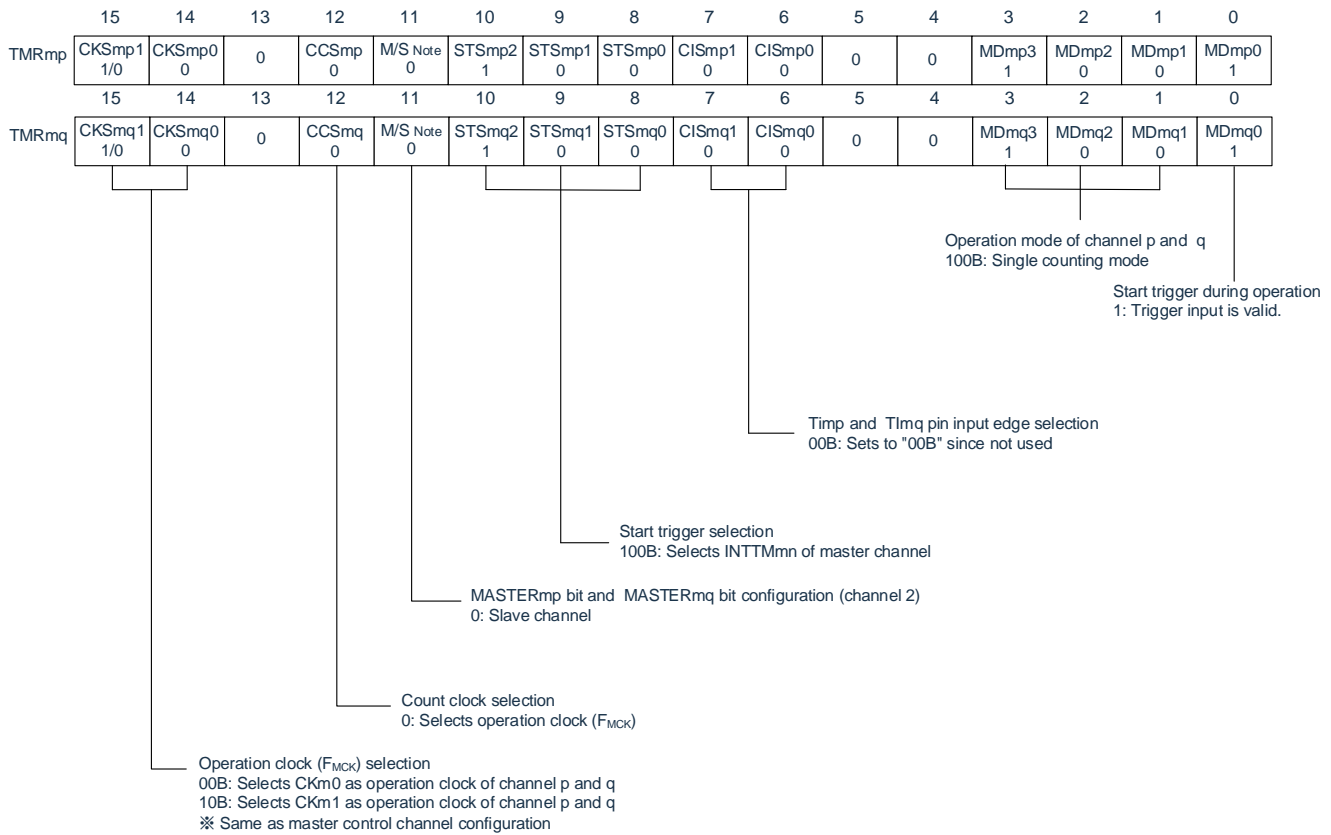
Note 1: m: unit number (m= 0, 1) n: master channel number (n=0)

Note 2: TMRm2: MASTERmn=1

TMRm0: Fixed to "0".

Figure 6-43: Example of register contents setting for multiple PWM output function (slave channel) (output two types of PWMs)

(a) Timer mode registers mp, mq (TMRmp, TMRmq)



(b) Timer output enable register m (TOEm)

| | bit q | bit p | |
|-----|-------|-------|-------------------------------------------------------------------|
| TOm | TOmq | TOmp | 0: utputs 0 from TOmp or TOmq. 1: Outputs 0 from TOmp or TOmq. |
| | 1/0 | 1/0 | |

(c) Timer output enable register m (TOEm)

| | bit q | bit p | |
|------|-------|-------|------------------------------------------------------------------------------------------------------------------------------------------|
| TOEm | TOEmq | TOEmp | 0: Stops the TOmp or TOmq output operation by counting operation. 1: Enables the TOmp or TOmq output operation by counting operation. |
| | 1/0 | 1/0 | |

(d) Timer output level register m (TOLm)

| | bit q | bit p | |
|------|-------|-------|---------------------------------------------------------------------------------------------|
| TOLm | TOLmq | TOLmp | 0: Positive logic output (active high level) 1: Negative logic output (active low level) |
| | 1/0 | 1/0 | |

(e) Timer output mode register m (TOMm)

| | bit q | bit p | |
|------|-------|-------|------------------------------------|
| TOMm | TOMmq | TOMmp | 1: Sets slave channel output mode. |
| | 1 | 1 | |

Note: m: unit number (m= 0) n: master channel number (n=0)

p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than n)

Table 6-36: Procedure for the multiple PWM output function (output two types of PWMs) (1/2)

| | Software operation | Hardware status |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer4 initial settings | | The input clock of timer unit m is in the stop-providing state. (Stop providing clock, cannot write to each register) |
| | Set the TM4mEN bit of the peripheral enable register 0(PER0) to "1". | The input clock of timer unit m is in the providing state and the channels are in the stop state. (Start providing clock, can write to each register) |
| | Set the timer clock selection register m (TPSm). Determine the clock frequency of CKm0 and CKm1. | |
| Initial setting of channels | Set the timer mode registers mn, mp, (TMRmn, TMRmp,) for each channel used (to determine the channel operation mode). Set the interval (period) value for the master channel's timer data register mn (TDRmn), and set the duty cycle | The channel is in the stop state. (Provides clock, and consumes some power) |
| | Slave channel setting Set TOMmp and TOMmq bits of the timer output mode register m (TOMm) to "1" (slave channel output mode). Set the TOLmp and TOLmq bits to "0". | The TOmp pin is in Hi-Z output state. When the port mode register is in output mode and the port register is "0", the TOmp and TOMq initial set levels are output. |
| | Set the TOmp and TOMq bits and determine the initial output level of the TOmp and TOMq bits. | The TOmp and TOMq remains unchanged because the channel is in the stop state. |
| | Set the TOEmp amd TOEmq bits to "1" and enable TOmp and TOMq output. | The TOmp pin and TOMq pin output the levels set by the TOmp and TOMq. |
| | Set the Port Register and Port Mode Register to "0". | |

Table 6-36: Procedure for the multiple PWM output function (output two types of PWMs) (2/2)

| | Software operation | Hardware status |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Restart operation | Start operation (Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits. | TEMn = 1, TEMp, TEMq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting. |
| | In operation Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSR0q registers are not used. | The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOMq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated. |
| | Stop operation The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits. | TEMn, TEMp, TEMq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOMq output are not initialized but hold current status. |
| | The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOMq bits. | The T0mp and T0mq pins output the T0mp and T0mq set levels. |
| | Timer4 stop To hold the TOmp and TOMq pin output levels Clears the TOmp and TOMq bits to 0 after the value to be held is set to the port register. When holding the TOmp and TOMq pin output levels are not necessary: <u>Setting not required</u> The TM4mEN bit of the PER0 register is cleared to 0. | The TOmp and TOMq pin output levels are held by port function. The input clock of timer unit m is in the stop-providing state. Initialize all circuits and the SFR for each channel. (TOmp bit and TOMq bit become "0" and TOmp pin and TOMq pin become port function) |

Note: m: unit number (m= 0) n: master channel number (n=0)

p: slave channel number q: slave channel number $n < p < q \leq 3$ (p and q are integers greater than

n)

Chapter 7 LSITIMER 12-Bit Interval Timer

7.1 Function of 12-bit interval timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from sleep mode, deep sleep mode and partial power-down mode.

7.2 Configuration of 12-bit interval timer

The 12-bit interval timer includes the following hardware.

Table 7-1: Configuration of 12-bit interval timer

| Item | Configuration |
|------------------|-----------------------------------------------|
| Counter | 12-bit counter |
| Control register | 12-bit interval timer control register (CON0) |

7.3 Register mapping

(CON0 base address = 0x4004_4B50)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------------------|-------------|
| CON0 | 0x000 | R/W | 12-bit interval timer control register | 0xFF |

7.4 12-bit interval timer control register (CON0)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The CON0 register can be set by a 12-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to “FFFH”.

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------------------------------------------------------|-------------|
| 15 | RINTE | 12-bit Interval timer operation control 0: Count operation stopped (count clear) 1: Count operation started | 0 |
| 14:12 | -- | Reserved | -- |
| 11:0 | ITCMP | Specification of the 12-bit interval timer compare value | 0xFFF |

The setting of the ITCMP compare value and the calculation of the related interrupt period are shown in the following table: (for reference only)

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| ITCMP[11]~ITCMP[0] | Specification of the 12-bit interval timer compare value |
| 001H | These bits generate a fixed-cycle interrupt of “Count Clock Cycle (ITCMP Set value + 1)”. |
| • | |
| • | |
| • | |
| FFFH | |
| 000H | Settings are prohibited. |
| Example interrupt cycles when 001H or FFFH is specified for ITCMP[11] ~ ITCMP[0] ITCMP[11]~ITCMP[0]=001H, count clock: Fclk=15kHz 1/15[kHz]*(1+1)=0.13333 [ms] ITCMP[11]~ITCMP[0]=FFFH, count clock: Fclk=15kHz 1/15[kHz]*(4095+1)=273.06667[ms] | |

Note 1: Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

Note 2: The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

Note 3: When setting the CON0 register after returned from normal operation mode and entering sleep mode again, confirm that the written value of the CON0 register is reflected, or wait that more than one clock of the count clock has elapsed. Then enter sleep mode.

Note 4: Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.

Note 5: However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

7.5 12-bit interval timer operation

7.5.1 12-bit interval timer operation timing

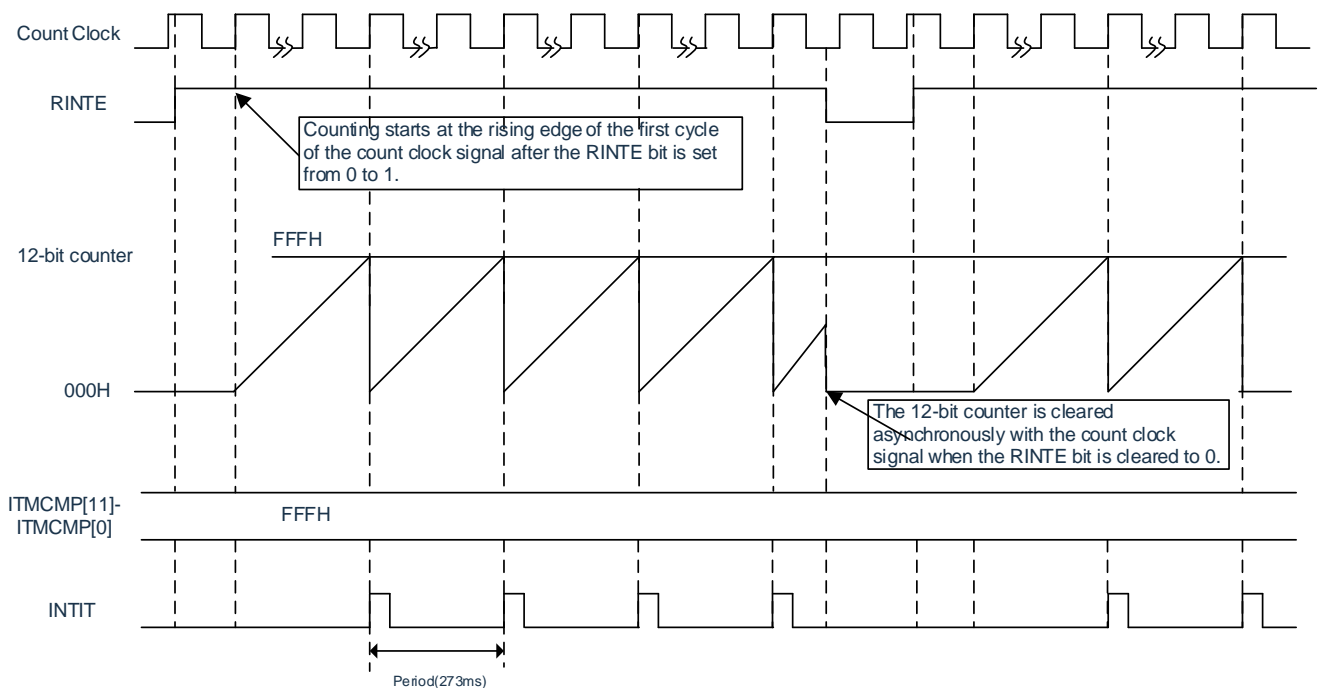
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate a 12-bit interval timer that repeatedly generates interrupt requests (INTIT). When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is shown in Figure 7-1.

Figure 7-1: 12-bit interval timer operation timing

(ITCMP[11]~ITCMP[0]=FFFH, count clock: Fclk=15KHz)

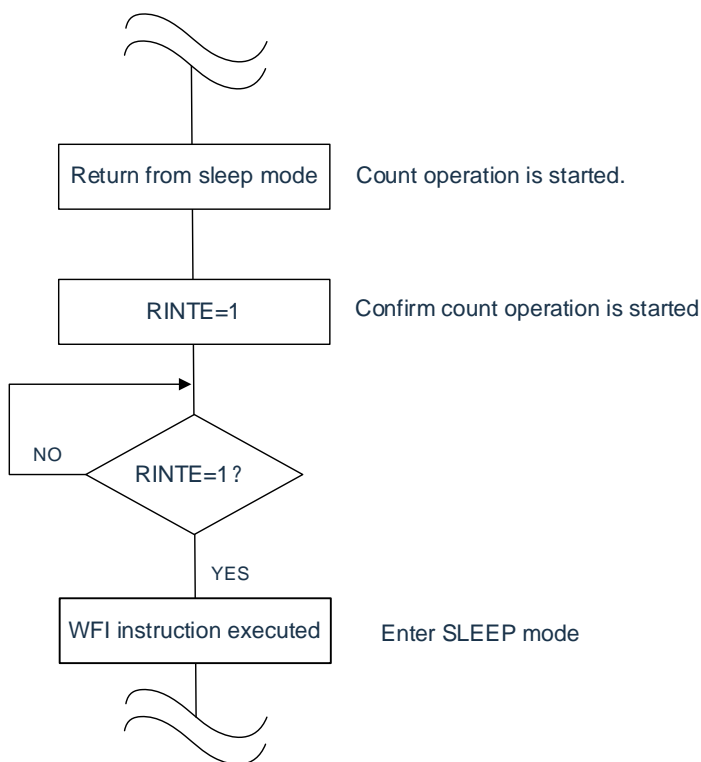


7.5.2 Start of count operation and re-enter to sleep mode after returned from sleep mode

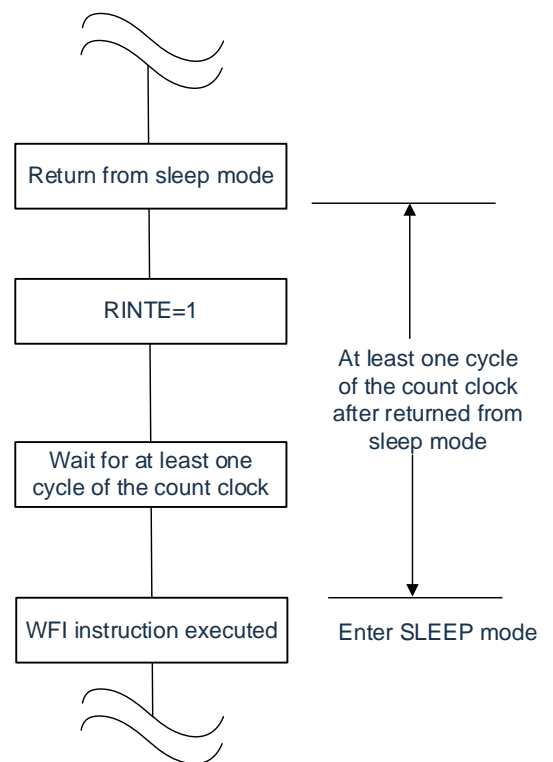
When setting the RINTE bit after returned from sleep mode and entering sleep mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock. Then, enter sleep mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter sleep mode (see Example 1 in the Figure below).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter sleep mode (see Example 2 in the Figure below).

Example 1



Example 2



Chapter 8 Clock Output/Buzzer Output Controller

8.1 Function of clock output/buzzer output controller

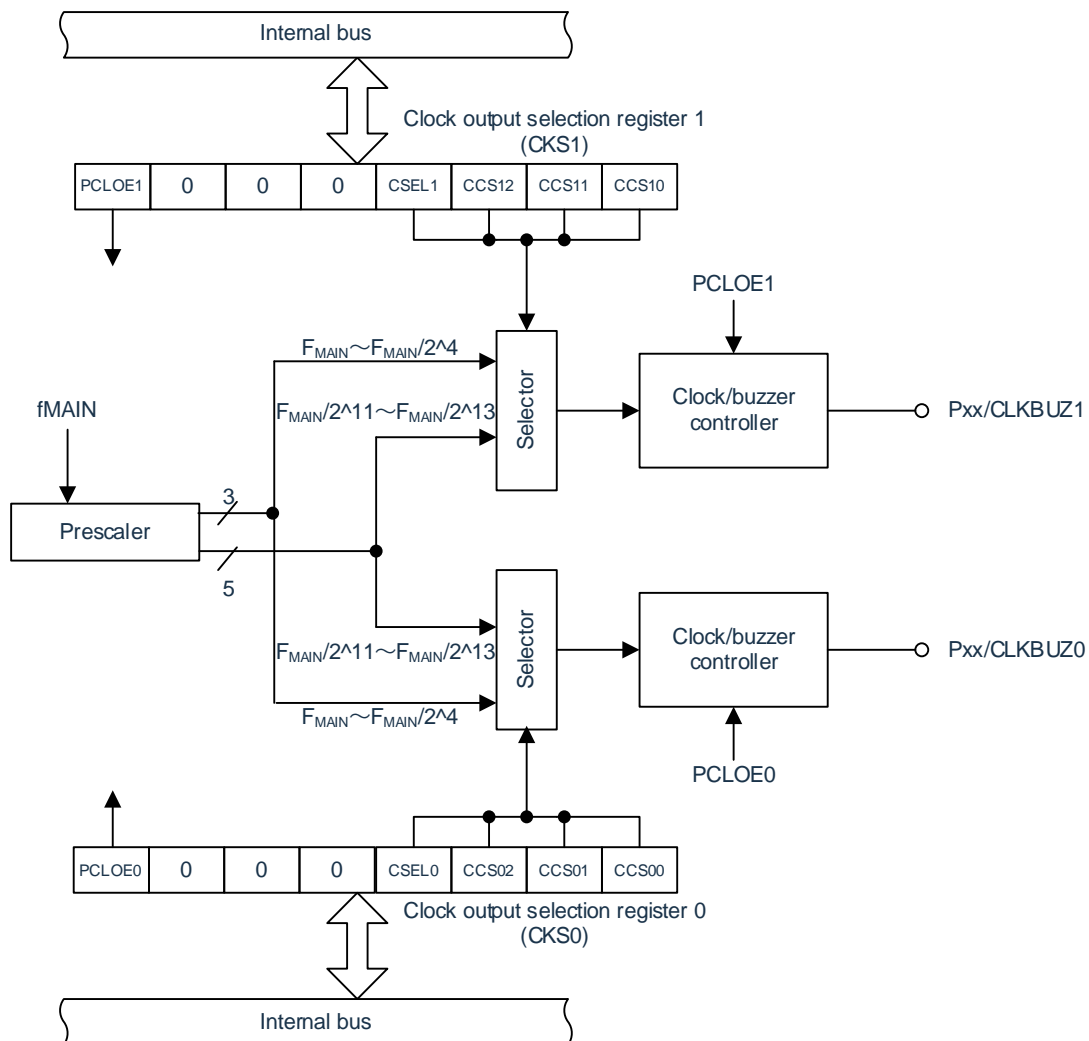
Clock output is the function of outputting the clock provided to the peripheral IC, and buzzer output is the function of outputting the buzzer frequency square wave.

This product has two clock output/buzzer output pins, CLKBUZ0 can be used as clock output or buzzer output from P20, P25, P00, P06, P22, and CLKBUZ1 can be used as clock output or buzzer output from P21, P25, P06, and P22.

The CLKBUZn pin outputs the clock selected by the clock output selection register n (CKSn).

The block diagram of the clock output/buzzer output controller is shown in Figure 8-1.

Figure 8-1: Block diagram of clock output/buzzer output controller



Note: For the frequencies that can be output from CLKBUZ0 and CLKBUZ1 pins, please refer to “AC Characteristics” in the data sheet.

8.2 Structure of clock output/buzzer output controller

The clock output/buzzer output controller consists of the following hardware.

Table 8-1: Structure of clock output/buzzer output controlle

| Item | Structure |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Control registers | Clock output select registers n (CKSn) Port mode control register (PMCxx), Port mode register (PMxx), Port multiplexing control register (PxxCFG) |

8.3 Register mapping

(CKS0/1 base address = 0x4004_0FA5)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|--------------------------------|-------------|
| CKS0 | 0x000 | R/W | Clock output select register 0 | 0x0 |
| CKS1 | 0x001 | R/W | Clock output select register 1 | 0x0 |

8.3.1 Clock output select register (CKS0)

The register sets output enable/disable for clock output or for the buzzer frequency output pin (CLKBUZn), and sets the output clock.

Select the clock to be output from the CLKBUZn pin by using the CKSn register. The CKSn register is set by an 8-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes "00H".

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------|-------------|
| 7 | PCLOE0 | CLKBUZ0 pin output enable/disable 0: Output disable (default) 1: Output enable | 0 |
| 6:4 | -- | Reserved | -- |
| 3 | CSEL0 | CLKBUZ0 pin output clock selection | 0 |
| 2:0 | CCS0 | CLKBUZ0 pin output clock selection | 0x0 |

The specific CLKBUZn pin output clocks are selected in the following table:

| CSEL0 | CCS0[2] | CCS0[1] | CCS0[0] | CLKBUZ0 pin output clock selection |
|-------|---------|---------|---------|------------------------------------|
| 0 | 0 | 0 | 0 | F_{MAIN} |
| 0 | 0 | 0 | 1 | $F_{MAIN}/2$ |
| 0 | 0 | 1 | 0 | $F_{MAIN}/2^2$ |
| 0 | 0 | 1 | 1 | $F_{MAIN}/2^3$ |
| 0 | 1 | 0 | 0 | $F_{MAIN}/2^4$ |
| 0 | 1 | 0 | 1 | $F_{MAIN}/2^{11}$ |
| 0 | 1 | 1 | 0 | $F_{MAIN}/2^{12}$ |
| 0 | 1 | 1 | 1 | $F_{MAIN}/2^{13}$ |

Note 1: Use the output clock within a range of 16 MHz. For details, please refer to "AC Characteristics" in the data sheet.

Note 2: Change the output clock after disabling clock output (PCLOE0 = 0).

Note 3: To shift to deep sleep mode when the main system clock is selected (CSEL0=0), set PCLOE0=0 before executing the WFI instruction.

Note 4: F_{MAIN} : Main system clock frequency

8.3.2 Clock output select register (CKS1)

The register sets output enable/disable for clock output or for the buzzer frequency output pin (CLKBUZ1), and sets the output clock.

Select the clock to be output from the CLKBUZ1 pin by using the CKS1 register. The CKS1 register is set by a 32-bit memory manipulation instruction. After a reset signal is generated, the value of this register becomes "00H".

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------|-------------|
| 7 | PCLOE1 | CLKBUZ1 pin output enable/disable 0: Output disable (default) 1: Output enable | 0 |
| 6:4 | -- | Reserved | -- |
| 3 | CSEL1 | CLKBUZ1 pin output clock selection | 0 |
| 2:0 | CCS1 | CLKBUZ1 pin output clock selection | 0x0 |

The specific CLKBUZn pin output clocks are selected in the following table:

| CSEL1 | CCS1[2] | CCS1[1] | CCS1[0] | CLKBUZ1 pin output clock selection |
|-------|---------|---------|---------|------------------------------------|
| 0 | 0 | 0 | 0 | F_{MAIN} |
| 0 | 0 | 0 | 1 | $F_{MAIN}/2$ |
| 0 | 0 | 1 | 0 | $F_{MAIN}/2^2$ |
| 0 | 0 | 1 | 1 | $F_{MAIN}/2^3$ |
| 0 | 1 | 0 | 0 | $F_{MAIN}/2^4$ |
| 0 | 1 | 0 | 1 | $F_{MAIN}/2^{11}$ |
| 0 | 1 | 1 | 0 | $F_{MAIN}/2^{12}$ |
| 0 | 1 | 1 | 1 | $F_{MAIN}/2^{13}$ |

Note 1: Use the output clock within a range of 16 MHz. For details, please refer to "AC Characteristics" in the data sheet.

Note 2: Change the output clock after disabling clock output (PCLOE1 = 0).

Note 3: To shift to deep sleep mode when the main system clock is selected (CSEL1=0), set PCLOE1 = 0 before executing the WFI instruction.

Note 4: F_{MAIN} : Main system clock frequency

8.4 Registers for configuring clock output/buzzer output port functions

This product has two clock output/buzzer output pins, CLKBUZ0 can be used as clock output or buzzer output from P20, P25, P00, P06, P22, and CLKBUZ1 can be used as clock output or buzzer output from P21, P25, P06, P22.

To use the clock output/buzzer output function, the port multiplexing function configuration register (PxxCFG), port register (Pxx), port mode register (PMxx), and port mode control register (PMCxx) must be set. For details, refer to “Chapter 3 Pin Function”.

A multiplexed port configured as a clock output/buzzer output pin must have its corresponding Port Register (Pxx), Port Mode Register (PMxx) bits and Port Mode Control Register (PMCxx) bits set to “0”.

(Example) Using P20 as clock output/buzzer output (CLKBUZ0):

Set bit P20 of the port register 2 to “0”.

Set bit PM20 of the port mode register 2 to “0”.

Set bit PMC20 of the port mode control register 2 to “0”.

Set bit P20CFG of the port multiplexing function configuration register to “0x01”.

(Example) Using P21 as clock output/buzzer output (CLKBUZ1):

Set bit P21 of the port register 2 to “0”.

Set bit PM15 of the port mode register 2 to “0”.

Set bit PMC15 of the port mode control register 2 to “0”.

Set bit P15CFG of the port multiplexing function configuration register to “0x01”.

8.5 Operation of clock output/buzzer output controller

One pin can be used as clock output or buzzer output.

The CLKBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

The CLKBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

8.5.1 Operation as output pin

The CLKBUZn pin is output as the following procedure:

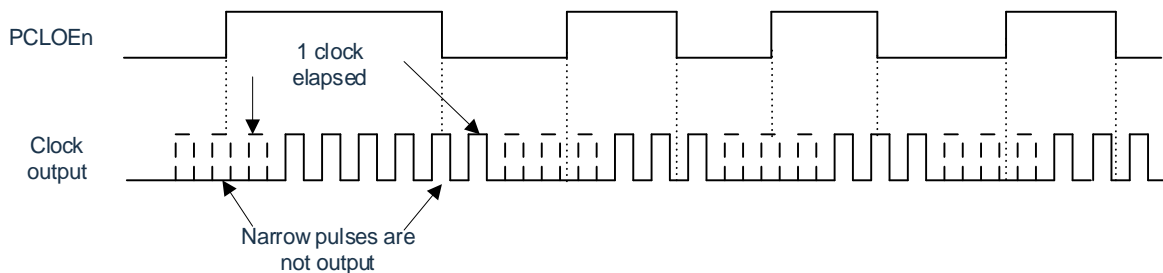
- 1) Set the port multiplexing function configuration register (PmnCFG). Set the bit of the port register (Pxx), port mode register (PMxx) and port mode control register (PMCxx) corresponding to the port used as CLKBUZ0 pin to "0".
- 2) Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the CLKBUZn pin (output in disabled status).
- 3) Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Note 1: The controller used as clock output starts or stops the clock output after 1 clock after the clock output (PCLOEn bit) is enabled or disabled. At this time, pulses with a narrow width are not output.

Figure 8-2 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Note 2: n=0, 1

Figure 8-2: CLKBUZn pin output clock timing



8.6 Cautions of clock output/buzzer output controller

When the main system clock is selected for the CLKBUZn output (CSELn=0), if deep sleep mode is entered within 1.5 clock cycles output from the CLKBUZn pin after the output is disabled (PCLOEn=0), the CLKBUZn output width becomes shorter.

Chapter 9 Watchdog Timer

9.1 Function of watchdog timer

The counting operation of the watchdog timer is set by the option byte (000C0H). The watchdog timer operates on the low-speed on-chip oscillator clock (F_{IL}).

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- (1) If the watchdog timer counter overflows
- (2) If data other than “ACH” is written to the WDTE register
- (3) If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see Chapter 24 Reset Function. When 75% of the overflow time + $1/2 F_{IL}$ is reached, an interval interrupt can be generated.

9.2 Configuration of watchdog timer

The watchdog timer includes the following hardware.

Table 9-1: Configuration of watchdog timer

| Item | Configuration |
|------------------|---------------------------------------|
| Counter | Internal counter (17 bits) |
| Control register | Watchdog timer enable register (WDTE) |

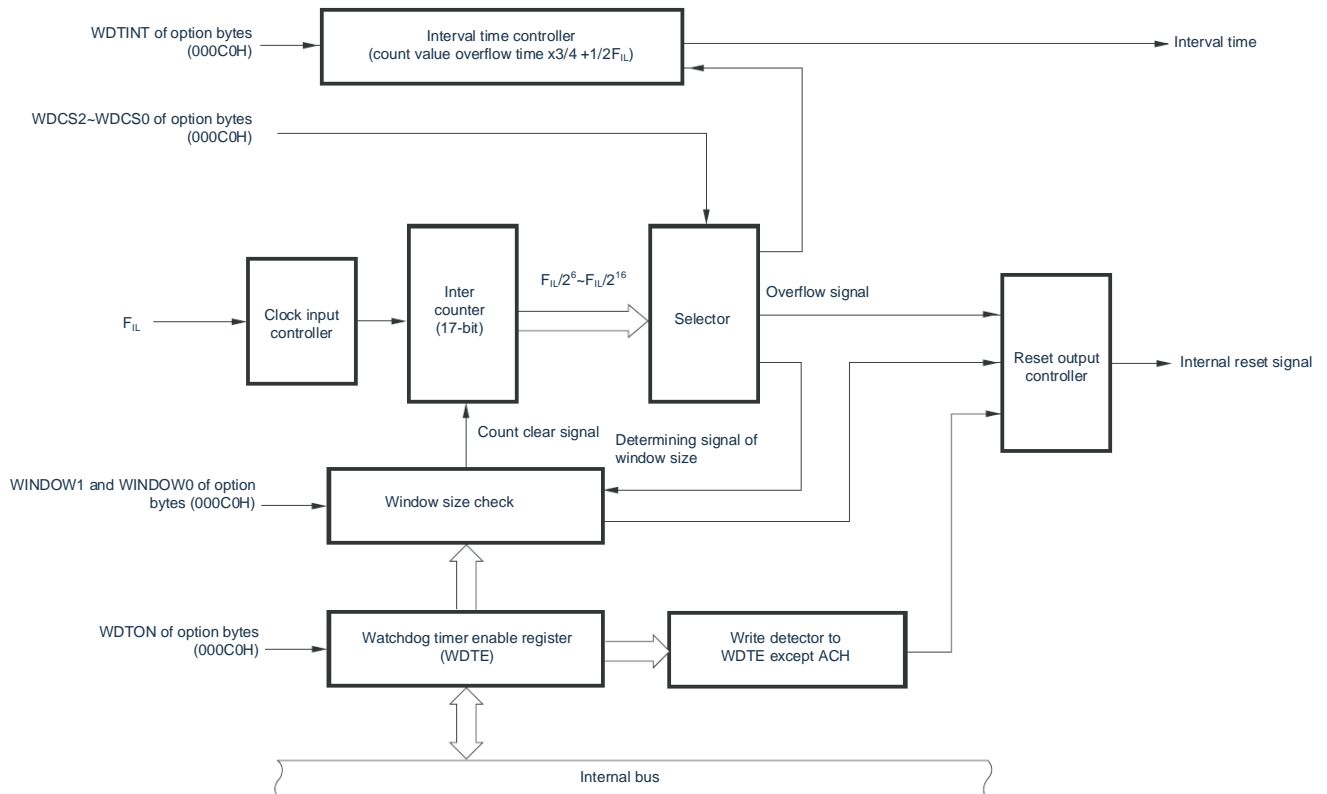
The operation of the counter is controlled by the option byte as well as the setting of the overflow time, the window opening period and the interval interrupt.

Table 9-2: Setting of option bytes and watchdog timer

| Setting of Watchdog Timer | Option Byte (000C0H) |
|-----------------------------------------------------------------|----------------------------------|
| Watchdog timer interval interrupt | bit7 (WDTINT) |
| Window open period | bit6 and bit5 (WINDOW1, WINDOW0) |
| Controlling counter operation of watchdog timer | bit4 (WDTON) |
| Overflow time of watchdog timer | bit3~1 (WDCS2~WDCS0) |
| Controlling counter operation of watchdog timer (in sleep mode) | bit0 (WDSTBYON) |

Note: For option byte, see Chapter 29 Option Byte.

Figure 9-1: Block diagram of watchdog timer



Note: F_{IL}: Low-speed on-chip oscillator clock frequency

9.3 Register mapping

(WDTE base address = 0x4002_1001)

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|--------------------------------|-------------|
| WDTE | 0x000 | R/W | Watchdog timer enable register | 0x1A/0x9A |

(LOCKCTL base address = 0x4002_0405)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------|-------------|
| LOCKCTL | 0x000 | R/W | Control register | 0x1 |

(PRCR base address = 0x4002_0406)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|---------------------|-------------|
| PRCR | 0x000 | R/W | Protection register | 0x0 |

9.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again. The WDTE register is set by an 8-bit memory manipulation instruction. Reset signal generation sets this register to “9AH” or “1AH” ^{Note}.

| Bit | Symbol | Description | Reset value |
|-----|--------|----------------------------------------------------------------------|-------------|
| 7:0 | WDTE | Write 0xAC to clear the watchdog timer counter and restart counting. | 0x1A/0x9A |

Note: The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

| WDTON bit setting value | WDTE register reset value |
|---------------------------------------------|---------------------------|
| 0 (watchdog timer count operation disabled) | 1AH |
| 1 (watchdog timer count operation enabled) | 9AH |

Note 1: If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Note 2: The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

9.3.2 LOCKUP control register (LOCKCTL)

The LOCKCTL register is a configuration register for controlling the Cortex-M0+ LockUp function to operate the watchdog timer, and PRCR is its write-protect register.

The LOCKCTL register is set by an 8-bit memory manipulation instruction.

After generating a reset signal, the value of the LOCKCTL register changes to "01H".

| Bit | Symbol | Description | Reset value |
|-----|------------|---------------------------------------------------------------------------------------------------------------|-------------|
| 7:1 | - | Reserved | - |
| 0 | lockup_rst | Configuration of LOCKUP function 0: LOCKUP does not cause a WDT reset 1: LOCKUP causes the WDT to reset | 1 |

9.3.3 Protection register (PRCR)

The LOCKCTL register is a configuration register for controlling the Cortex-M0+ LockUp function to operate the watchdog timer, and PRCR is its write-protect register.

The PRCR register is set by an 8-bit memory manipulation instruction.

After generating a reset signal, the value of the PRCR register changes to "00H".

| Bit | Symbol | Description | Reset value |
|-----|--------|-----------------------------------------------------------------------------------------------------|-------------|
| 7:1 | PRTKEY | Write protection of PRCR 78H: PRCR is writable Other: PRCR is not writable | 0x0 |
| 0 | PRCR | Write protection of LOCKUP control register 0: LOCKCTL is not writable 1: LOCKCTL is writable | 0 |

9.3.4 Watchdog configuration register (WDTCFG0/1/2/3)

The WDTCFGx configuration register is a register that forces the watchdog timer to operate or not.

The WDTCFGx register is set by an 8-bit register manipulation instruction.

After a reset signal is generated, the value of the WDTCFGx register changes to “00H”.

WDTCFGx configuration register

| Bit | Symbol | Description | Reset value |
|-----|---------|-----------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:0 | WDTCFGx | The Watchdog Configuration Register can be set to a specific value to force the watchdog timer to run. See the table below. | 0x0 |

| WDTCFG0 | WDTCFG1 | WDTCFG2 | WDTCFG3 | Configuration of the watchdog timer function |
|---------|---------|---------|---------|---------------------------------------------------------------------------------------------------|
| 0x1A | 0x2B | 0x3C | 0x4D | The operation of the watchdog timer after reset is determined by the option byte ^{Note1} |
| Other | | | | Forces the watchdog timer to run after reset |

Note 1: For detailed configuration refer to section 29.4 User Option Byte

9.4 Operation of watchdog timer

9.4.1 Operational control of watchdog timer

1. When using the watchdog timer, set the following items by option byte (000C0H):

- (1) The bit 4 (WDTON) of the option byte (000C0H) must be set to "1" to enable the watchdog timer count to operate (the counter starts operating after the reset is released) (refer to Chapter 29 Option Byte for details).

| WDTON | Counter of watchdog timer |
|-------|------------------------------------------------------------------|
| 0 | Disables counting operation (stop counting after reset released) |
| 1 | Enables counting operation (start counting after release reset) |

- (2) The overflow time must be set by bit3~1 (WDCS2~WDCS0) of the option byte (000C0H) (refer to 9.4.2 and Chapter 29 Option Byte for details).

- (3) The window opening period must be set by bit6 and bit5 (WINDOW1, WINDOW0) of the option byte (000C0H) (refer to 9.4.2 and Chapter 29 Option Byte for details).

2. After the reset is released, the watchdog timer starts counting.

3. After starting counting and before the overflow time set by the option byte, writing "ACH" to the watchdog timer enable register (WDTE) clears the watchdog timer and starts counting again.

4. Thereafter, writes to WDTE registers after the second time after the reset must be performed while the window is open. If you write the WDTE register while the window is closed, an internal reset signal is generated.

5. If you do not write "ACH" to the WDTE register and exceed the overflow time, an internal reset signal is generated. An internal reset signal is generated if:

- (1) If data other than "ACH" is written to the WDTE register

Note 1: When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Note 2: After "ACH" is written to the WDTE register, an error of up to 2 F_{IL} clocks may occur before the watchdog timer is cleared.

Note 3: The watchdog timer can be cleared immediately before the count value overflows.

Note 4: As shown below, the watchdog timer operates in sleep or deep sleep mode depending on the set value of bit0 (WDSTBYON) of the option byte (000C0H).

| | WDSTBYON=0 | WDSTBYON=1 |
|-----------------|-----------------------------------|---------------------------------------|
| Sleep mode | Stop operation of watchdog timer. | Continue operation of watchdog timer. |
| Deep sleep mode | | |

When the WDSTBYON bit is "0", restart the watchdog timer count after the sleep or deep sleep mode released. At this point, the counter is cleared to "0" and the count begins.

If the period between the deep sleep mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.

9.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time. The following overflow times can be set.

Table 9-3: Setting of overflow time of watchdog timer

| WDCS2 | WDCS1 | WDCS0 | Overflow time of watchdog timer (When $F_{IL}=20\text{kHz}(\text{MAX.})$) |
|-------|-------|-------|-------------------------------------------------------------------------------|
| 0 | 0 | 0 | $2^6/F_{IL}$ (3.2ms) |
| 0 | 0 | 1 | $2^7/F_{IL}$ (6.4ms) |
| 0 | 1 | 0 | $2^8/F_{IL}$ (12.8ms) |
| 0 | 1 | 1 | $2^9/F_{IL}$ (25.6ms) |
| 1 | 0 | 0 | $2^{11}/F_{IL}$ (102.4ms) |
| 1 | 0 | 1 | $2^{13}/F_{IL}$ (409.6ms) |
| 1 | 1 | 0 | $2^{14}/F_{IL}$ (819.2ms) |
| 1 | 1 | 1 | $2^{16}/F_{IL}$ (3276.8ms) |

Note: F_{IL} : Low-speed on-chip oscillator clock frequency

9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows:

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Note: When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 9-4: Setting window open period of watchdog timer

| WINDOW1 | WINDOW0 | Window open period of watchdog timer |
|---------|---------|--------------------------------------|
| 0 | - | Settings are disabled. |
| 1 | 0 | 75% |
| 1 | 1 | 100% |

Note 1: When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Note 2: If the overflow time is set to $2^9/F_{IL}$, the window close time and open time are as follows.

| | Setting of window open period | |
|-------------------|-------------------------------|----------|
| | 75% | 100% |
| Window close time | 0~12.8ms | None |
| Window open time | 12.8~25.6ms | 0~25.6ms |

<When window open period is 75%>

(1) Overflow time:

$$2^9/F_{IL}(\text{MAX.})=2^9/20\text{kHz}(\text{MAX.})=25.6\text{ms}$$

(2) Window close time:

$$0\sim 2^9/F_{IL}(\text{MIN.})\times(1-0.75)=0\sim 2^9/10\text{kHz}\times 0.25=0\sim 12.8\text{ms}$$

(3) Window open time:

$$2^9/F_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/F_{IL}(\text{MAX.})=12.8\sim 25.6\text{ms}$$

9.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when $75\% + 1/2F_{IL}$ of the overflow time is reached.

Table 9-5: Setting of watchdog timer interval interrupt

| WDTINT | Use of watchdog timer interval interrupt |
|--------|------------------------------------------------------------------------------------------|
| 0 | Interval interrupt is not used. |
| 1 | Interval interrupt is generated when $75\% + 1/2F_{IL}$ of the overflow time is reached. |

Note 1: When operating with the X1 oscillation clock after releasing the deep sleep mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the deep sleep mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.

Note 2: The watchdog timer continues counting even after INTWDTI is generated (until “ACH” is written to the watchdog timer enable register (WDTE)). If “ACH” is not written to the WDTE register before the overflow time, an internal reset signal is generated.

9.4.5 Operation of watchdog timer during LOCKUP

When lockup_rst bit of the lockup control register lockcTL is set to 1, once the kernel enters the LOCKUP state, the low-speed internal oscillator begins to oscillate, the watchdog timer automatically starts operating, and the overflow time control bit (WDCS2~WDCS0) is set to 3'b010, that is, the overflow time is set to 12.8ms.

Chapter 10 DIVSQRT Unit

10.1 Overview

The chip contains a 32-bit/32-bit hardware divider and a 32-bit hardware square root extractor.

10.2 Features

- ◆ Support signed/unsigned division and square root operations.
- ◆ Both quotient and remainder are 32 bits wide.
- ◆ Divider has a clear flag indicator bit.
- ◆ Operations are completed within 22 APB clocks.
- ◆ Start the operation by writing to the ALUB register.

10.3 Functional description

The operation unit can choose between division mode and square root mode through the register DIVSQRT->CON[4]. In division mode, the quotient is saved in DIVSQRT->RES0 and the remainder is saved in DIVSQRT->RES1. The DIVSQRT->CON[2] register can be used to detect whether the divisor is zero, which is a read-only bit. In square root mode, DIVSQRT->RES0 saves the square root result, and DIVSQRT->RES1 is unused.

Note that in square root mode, if the highest bit of the number being square rooted is 1, it is treated as a signed number. The absolute value is taken first before performing the square root operation:

$$RES0 = \sqrt[3]{absval(ALUB)}$$

The register DIVSQRT->CON[3] can be used to detect whether the operation has been completed. This is a read-only bit, with a value of 0 indicating that the operation is still ongoing and a value of 1 indicating that the operation has been completed. When the division unit is idle, this bit is also set to 1.

The register DIVSQRT->CON[1] can be used to select between signed and unsigned division modes.

Please note that the clock enable bit for the arithmetic unit is set in the peripheral enable register PER12.

Note: Do not write to the ALUA or ALUB registers or read from the RES0 or RES1 registers during calculation, otherwise the results are unpredictable.

Register definitions in different modes:

| Operation unit mode | ALUA | ALUB | RES0 | RES1 |
|---------------------|----------|----------|----------------------------------------|-----------|
| Division mode | Dividend | Divisor | Quotient | Remainder |
| Square root mode | - | Radicand | Square root result (low 16 bits valid) | - |

10.4 Register mapping

(DIVSQRT base address = 0x4006_4380)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------------|-------------|
| CON | 0x000 | R/W | Operation unit control register | 0xC |
| ALUA | 0x004 | R/W | Operation unit data A register | 0x0 |
| ALUB | 0x008 | R/W | Operation unit data B register | 0x0 |
| RES0 | 0x00C | RO | Operation unit result 0 register | 0x0 |
| RES1 | 0x010 | RO | Operation unit result 1 register | 0x0 |

10.5 Register description

10.5.1 DIVSQRT control register (DIVSQRTCON)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:5 | - | Reserved | - |
| 4 | MODE | Operation mode select bit 0: Division mode 1: Square root mode | 0 |
| 3 | READY | Operation completion indicator bit 0: Operation is ongoing 1: Operation is completed or is in idle state | 1 |
| 2 | DIVBY0 | Division mode clear indicator bit (this bit is updated automatically when the divisor is written) 0: Divisor is not 0 1: Divisor is 0 | 1 |
| 1 | SIGN | Division mode dymbol select bit 0: Unsigned mode 1: Signed mode | 0 |
| 0 | - | Reserved | 0 |

10.5.2 DIVSQRT data A register (DIVSQRTALUA)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------|-------------|
| 31:0 | ALUA | 32-bit data A | 0x0 |

10.5.3 DIVSQRT data B register (DIVSQRTALUB)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------|-------------|
| 31:0 | ALUB | 32-bit data B | 0x0 |

10.5.4 DIVSQRT result 0 register (DIVSQRTRES0)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------|-------------|
| 31:0 | RES0 | 32-bit result 0 | 0x0 |

10.5.5 DIVSQRT result 1 register (DIVSQRTRES1)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------|-------------|
| 31:0 | RES1 | 32-bit result 1 | 0x0 |

Chapter 11 Timer (TIMER0/1)

11.1 Overview

It contains two programmable 32-bit/16-bit counters, TIMER0/TIMER1, providing users with convenient timer counting functions.

11.2 Features

- ◆ Configurable 32-bit/16-bit count down counter.
- ◆ Each timer has an independent prescaler.
- ◆ Support single trigger, periodic count, and continuous count modes.
- ◆ Support waking up the chip from sleep mode.

11.3 Function description

11.3.1 Single trigger mode

In single trigger mode, when the timer is enabled, the counter loads the initial value from the load register, counts down, and stops working when the counter reaches 0. An interrupt is generated at this time. To start the single trigger mode again, the TMROS bit needs to be cleared and then set again.

(When starting the single trigger mode again, it should be noted that the time during which the TMROS bit is cleared and remains 0 should be greater than one timer counting cycle.)

11.3.2 Periodic count mode

In periodic count mode, when the timer is enabled, the counter loads the initial value from the load register, counts down, and when the counter reaches 0, it loads the initial value again from the load register and continues counting. An interrupt is generated at this time.

11.3.3 Continuous count mode

In continuous count mode, when the timer is enabled, the counter loads the initial value from the load register, counts down, and when the counter reaches 0, it loads the maximum value as the initial value and continues counting. An interrupt is generated at this time.

11.3.4 Delayed load function

When data is written to the load register, the counter does not continue to decrement but loads the initial value from the load register on the next `TIMER_CLK` rising edge and then starts decrementing.

When data is written to the delayed load register, the data is written into the load register on the next `TIMER_CLK` rising edge. If the counter has already started counting, it will wait until the current period count reaches 0 before loading the initial value from the load register.

11.4 Register mapping

(Timer0 base address= 0x4006_1000, Timer1 base address= 0x4006_1100)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-----------------------------------------|-------------|
| CON | 0x000 | R/W | Timer control register | 0x20 |
| LOAD | 0x004 | R/W | Timer load register | 0x0 |
| VAL | 0x008 | RO | Timer current value register | 0xFFFFFFFF |
| RIS | 0x00C | RO | Timer interrupt source status register | 0x0 |
| MIS | 0x010 | RO | Timer enabled interrupt status register | 0x0 |
| ICLR | 0x014 | WO | Timer interrupt clear register | - |
| BGLOAD | 0x018 | R/W | Timer delayed load register | 0x0 |

11.5 Register description

11.5.1 Timer control register (CON0/1)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7 | TMREN | Timer enable bit 0: Disable 1: Enable | 0 |
| 6 | TMRMS | Timer mode select bit 0: Continuous counting mode 1: Periodic counting mode | 0 |
| 5 | TMRIE | Timer interrupt enable bit 0: Disable interrupts 1: Enable interrupts | 1 |
| 4 | - | Reserved | - |
| 3:2 | TMRPRE | Timer prescaler 00: Divided by 1 01: Divided by 16 10: Divided by 256 11: Reserved | 0x0 |
| 1 | TMRSZ | Timer count bit selection 0: 16-bit counter 1: 32-bit counter | 0 |
| 0 | TMROS | Single trigger mode select bit 0: The mode is determined by the TMRMS bit 1: Single trigger mode (Re-triggered in one-shot mode, and the initial value of which is determined by the TMRMS bit) | 0 |

11.5.2 Timer load register (LOAD0/1)

| Bit | Symbol | Description | Reset value |
|------|----------|---------------------|-------------|
| 31:0 | TMRxLOAD | Timer load register | 0x0 |

11.5.3 Timer current value register (VAL0/1)

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------|-------------|
| 31:0 | TMRxVAL | Timer current count value | 0xFFFFFFFF |

11.5.4 Timer interrupt source status register (RIS0/1)

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------------------------------------------------------------------------|-------------|
| 31:1 | - | Reserved | - |
| 0 | TMRxRIS | Timer interrupt source status 1: An interrupt is generated 0: No interrupts generated | 0 |

11.5.5 Timer enabled interrupt status register (MIS0/1)

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:1 | - | Reserved | - |
| 0 | TMRxMIS | Timer has enabled the interrupt status bit 1: Interrupt enable and generate an interrupt 0: No interrupts generated | 0 |

11.5.6 Timer interrupt clear register (ICLR0/1)

| Bit | Symbol | Description | Reset value |
|------|----------|-----------------------------------------|-------------|
| 31:0 | TMRxICLR | Write any number, clear timer interrupt | - |

11.5.7 Timer delayed load register (BGLOAD0/1)

| Bit | Symbol | Description | Reset value |
|------|------------|----------------------------------------------------------------------------------------------------|-------------|
| 31:0 | TMRxBGLOAD | Timer delayed load register (reads the value of the most recent write to TMRxLOAD or TIMERxBGLOAD) | 0x0 |

Chapter 12 Capture/Compare/PWM Module (CCP0/1)

12.1 Overview

It contains 2 sets of CCP modules (CCP0/CCP1), each set of CCP corresponds to two channels A and B. CCP0 corresponds to CCP0A/CCP0B, CCP1 corresponds to CCP1A/CCP1B.

12.2 Features

- ◆ Up to 2 groups of CCP, supporting a maximum of 4 PWM outputs.
- ◆ Each group of CCP can have independent periods.
- ◆ CCPn has an internal 16-bit counter and can generate compare/overflow interrupts.
- ◆ CCPn has independent capture functionality, with the option to input signals on either A or B pins.
- ◆ CCP1 has 4-channel capture functionality, capable of simultaneously capturing CCP0A/CCP0B/CCP1A/CCP1B input signals.
- ◆ Capture mode 1 supports the reload of the CCP0 counter during capture operation.
- ◆ Internal channel CAP3 supports analog comparator output capture functionality.
- ◆ Internal channels CAP0-CAP3 support software capture functionality.



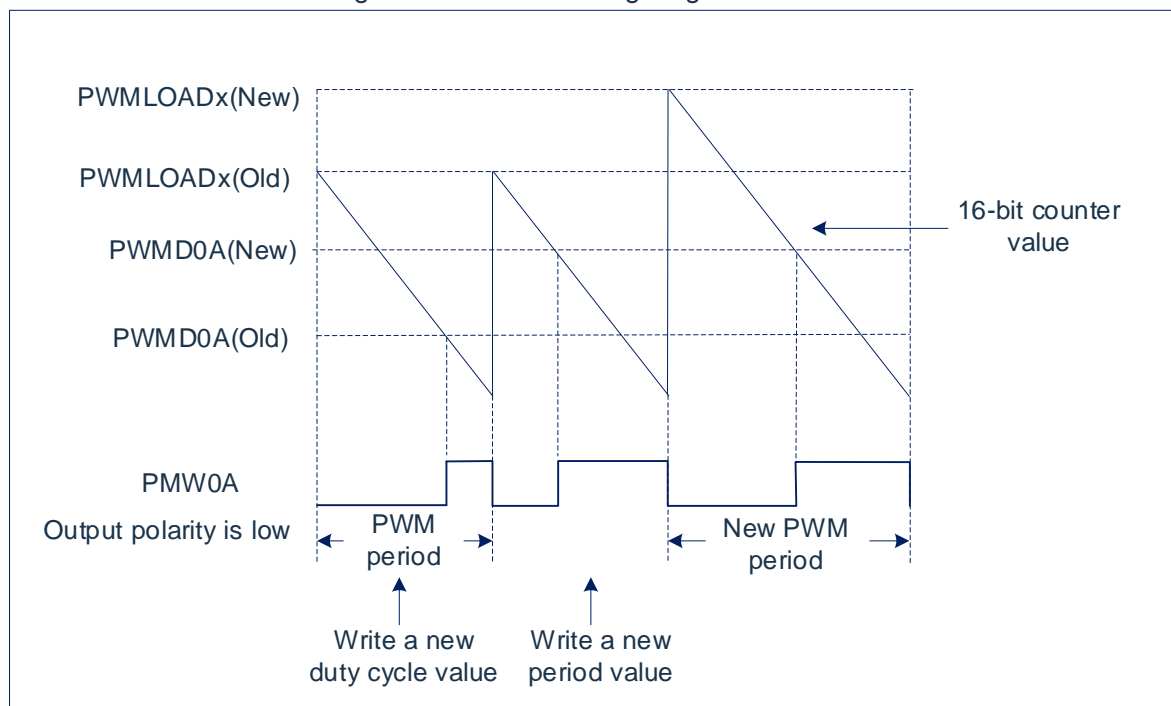
12.3 Function description

12.3.1 Pulse Width Modulation (PWM)

Each CCP can output two PWMs: PWMxA and PWMxB, which share one cycle, and the output duty cycle can be set independently by CCPDxA and CCPDxB. The polarity of PWMxA/PWMxB outputs can be set by PWMxAO/PWMxB0 bits, and correspond to CCPxA/CCPxB channel outputs respectively.

After set CCPx operation bit to 1, the 16-bit counter loads the value of CCPx reload register, counts down, and when the count value equals to the value of CCPDxA/B, the PWMxA/PWMxB output level changes.

Figure 12-1: PWM timing diagram



The calculation methods for period and duty cycle are as follows:

Period = $CCPLOADx \times CCP$ clock period.

PWMxA duty cycle = $CCPDxA / CCPLOADx$ (supports 0% to 100%).

PWMxB duty cycle = $CCPDxB / CCPLOADx$ (supports 0% to 100%).

When $CCPLOADx = 0$, the duty cycle for PWMxA and PWMxB is 0%.

When $CCPDxA / CCPDxB > CCPLOADx$, the duty cycle is 100%.

12.3.2 Square wave output mode

Square wave output mode is a type of pulse width modulation mode, where the period can be freely adjusted and the duty cycle is fixed at 50%.

Each group of CCP can be set to output mode from either A or B as BUZ. When setting the square wave output mode, PWM mode needs to be configured, and either CCPxCON.ZAEN or CCPxCON.ZAEN needs to be set. The period is configured by CCPLOADx, and the duty cycle is loaded with half of the CCPLOADx value by default (with the lowest bit ignored).



12.3.3 Capture mode 0

This capture mode is an external capture mode.

Each group of CCP can be set to use either channel A or channel B as the external capture signal pin. After setting CCPRUNx, the 16-bit counter counts down from 0xFFFF. When the capture condition is triggered, the counter stops counting, and CCPxA or CCPxB returns the current value of the counter. To perform the next capture, CCPRUNx needs to be cleared and then set again.

The calculation method for capture time is as follows:

CCPLOADx.RELOAD=0, capture time = $(0xFFFF - CCPDxA/B) \times CCPx \text{ clock period}$

CCPLOADx.RELOAD=1, capture time = $(CCPxLOAD[15:0] - CCPDxA/B) \times CCPx \text{ clock period}$.

12.3.4 Capture mode 1

CCP1 consists of 4 internal channels: CAP0, CAP1, CAP2, CAP3. One channel can select any one of the external channels in ECAP00-02 or ECAP10-13 as the capture channel. Alternatively, CCP0A, CCP0B, CCP1A, or CCP1B can be selected as separate capture channels.

ECAP00-02 corresponds to the positive inputs C0P0-C0P2 of analog comparator 0.

ECAP10-13 corresponds to positive inputs C1P0-C1P3 of analog comparator 1.

When using ECAP for external capture, the corresponding ports need to be set as the GPIO function.

When using CCP0A/CCP0B/CCP1A/CCP1B for capture, the corresponding ports need to be set as CCP ports.

Correspondence between CAPn and external channels:

| Internal channel | External channel | |
|------------------|----------------------|-----------------------------------------------|
| CAP0 | CAP0CHS=n: | Select ECAP0n (n=0-2)@ECAPS=0 |
| | CAP0CHS=n: | Select ECAP1n (n=0-3)@ECAPS=1 |
| | CAP0CHS=F: | Select CAP0A |
| | CAP0CHS=other value: | Reserved |
| CAP1 | CAP1CHS=n: | Select ECAP0n (n=0-2)@ECAPS=0 |
| | CAP1CHS=n: | Select ECAP1n (n=0-3)@ECAPS=1 |
| | CAP1CHS=F: | Select CAP0B |
| | CAP1CHS=other value: | Reserved |
| CAP2 | CAP2CHS=n: | Select ECAP0n (n=0-2)@ECAPS=0 |
| | CAP2CHS=n: | Select ECAP1n (n=0-3)@ECAPS=1 |
| | CAP2CHS=F: | Select CAP1A |
| | CAP2CHS=other value: | Reserved |
| CAP3 | CAP3CHS=n: | Select ECAP0n (n=0-2)@ECAPS=0 |
| | CAP3CHS=n: | Select ECAP1n (n=0-3)@ECAPS=1 |
| | CAP3CHS=8: | Select to output after ACMP0 filter selection |
| | CAP3CHS=9: | Select to output after ACMP1 filter selection |
| | CAP3CHS=F: | Select CAP1B |
| | CAP3CHS=other value: | Reserved |

12.3.5 Capture mode 2

This capture mode is external capture and is mainly used to capture PWM waveform information input from external sources.

In capture mode 2, channels CAP2 and CAP3 are mapped to CAP1, meaning that CAP1-3 are the same capture channel, and CAP0 is disabled.

CCP0 can be freely configured and is not affected.

CCP1 operates in counting mode, and after CCPRUN1 is set, the 16-bit counter counts down from the initial value. If an overflow occurs without a capture operation, the counter reloads the initial value and continues counting down.

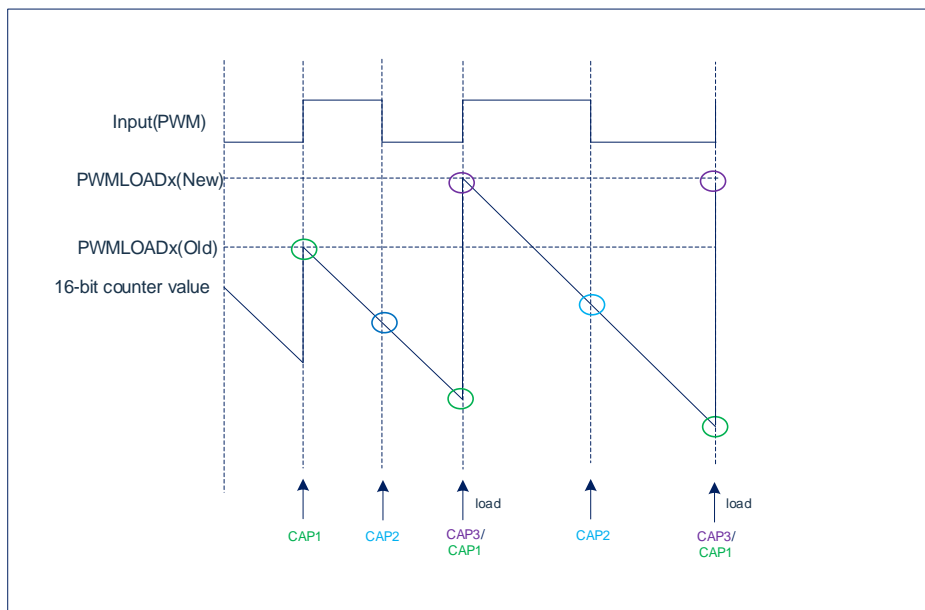
The operation for starting capture mode 2 is as follows:

- (1) After CAP1 triggers a capture, the CCP1 counter reloads the initial value and simultaneously loads it into CAP1DATA, allowing CAP2 to trigger a capture.
- (2) After CAP2 triggers a capture, the captured counter value is loaded into CAP2DATA, allowing CAP3 to trigger a capture.
- (3) After CAP3 triggers a capture, the captured counter value is loaded into CAP3DATA, completing the entire capture process. At this point, the values of CAP1DATA-CAP3DATA are loaded into CAP0DATA[31:16], and the values of CAP1DATA-CAP2DATA are loaded into CAP0DATA[15:0]. Then, CAP1 triggers a capture and the process returns to step (1).

Note:

- a: The first capture operation of CAP1 will also update the value of CAP0DATA, but this value has no reference value and should be discarded.
- b: If the time interval between completing the capture triggered by CAP3 exceeds one cycle of CCP1's counter, an overflow will occur, and the calculated value of CAP0DATA will be inaccurate. It is recommended to set the counter period far greater than the required PWM period.
- c: The time interval between the capture triggered by CAP1 to CAP2 and between CAP2 to CAP3 needs to be greater than 8 CCP1 counting values.
- d: Capture mode 2 supports software capture actions on CAP1-3.

e: CAP1-3 share the same capture channel, and capture operations will simultaneously generate corresponding capture flags on CAP1-CAP3.



12.3.6 PWM configuration process

- Configure PWM control registers by setting the prescaler, selecting the PWM mode, and enabling PWM.
- Configure the PWM period by writing to the CCPLOADx register.
- Configure the PWM duty cycle by writing to the CCPDxA/CCPDxB registers.
- If interrupts are required, enable the relevant interrupt bits and clear the interrupt status register.
- Set the corresponding I/O port as a PWM output.
- Set the PWM operation register to start the output.

12.3.7 Interrupt

In PWM mode, CCPx can generate two types of interrupts:

- Overflow interrupt: Generated when the counter decreases to 0.
- Compare interrupt: Generated when the counter value is equal to the value of CCPDxA or CCPDxB.

In capture mode 0/1, two types of interrupts can be generated:

- Overflow interrupt: Generated when the counter decreases to 0.
- Capture interrupt: Generated when the capture condition is triggered.



12.4 Register mapping

(CCP base address = 0x4006_4280) RO: read only; WO: write only; R/W: read/write.

| Register | Offset value | R/W | Description | Reset value |
|---------------------------|--------------|-----|---------------------------------------|-------------|
| CCPCON0 _(P1B) | 0x000 | R/W | CCP0 Control Register | 0x0 |
| CCPLOAD0 _(P1A) | 0x004 | R/W | CCP0 Reload Register | 0x0 |
| CCPD0A _(P1A) | 0x008 | R/W | CCP0 Channel A Data Register | 0x0 |
| CCPD0B _(P1A) | 0x00C | R/W | CCP0 Channel B Data Register | 0x0 |
| CCPCON1 _(P1B) | 0x010 | R/W | CCP1 Control Register | 0x0 |
| CCPLOAD1 _(P1A) | 0x014 | R/W | CCP1 Reload Register | 0x0 |
| CCPD1A _(P1A) | 0x018 | R/W | CCP1 Channel A Data Register | 0x0 |
| CCPD1B _(P1A) | 0x01C | R/W | CCP1 Channel B Data Register | 0x0 |
| CCPIMSC _(P1B) | 0x040 | R/W | CCP Interrupt Enable Register | 0x0 |
| CCPRIS | 0x044 | RO | CCP Interrupt Source Status Register | 0x0 |
| CCPMIS | 0x048 | RO | CCP Enabled Interrupt Status Register | 0x0 |
| CCPICLR | 0x04C | WO | CCP Interrupt Clear Register | 0x0 |
| CCPRUN _(P1B) | 0x050 | R/W | CCP Operation Register | 0x0 |
| CCPLOCK | 0x054 | R/W | CCP0/1 Write Enable Register | 0x0 |
| CAPCON _(P1B) | 0x058 | R/W | Capture Control Register | 0x0 |
| CAPCHS _(P1B) | 0x05C | R/W | Capture Channel Select Register | 0x0 |
| CAP0DAT0 _(P1A) | 0x060 | R/W | Capture Channel 0 Data Register | 0x0 |
| CAP1DAT0 _(P1A) | 0x064 | R/W | Capture Channel 1 Data Register | 0x0 |
| CAP2DAT0 _(P1A) | 0x068 | R/W | Capture Channel 2 Data Register | 0x0 |
| CAP3DAT0 _(P1A) | 0x06C | R/W | Capture Channel 3 Data Register | 0x0 |

Note:

The registers labeled with (P1A/P1B) are protected registers.

(P1A): When LOCK==55H or AAH, the labeled register is allowed to be written; =other values, write is forbidden.

(P1B): When LOCK==55H, the marked register is allowed to be written; =other values, write is forbidden.



12.5 Register description

12.5.1 CCPx control register (CCPCONx) (x=0,1)

| Bit | Symbol | Description | Reset value |
|-------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:10 | - | Reserved | - |
| 9 | CCPxZBEN | Square wave mode enable bit for channel B (valid for PWM mode) 0: Disable 1: Enable, and duty cycle loaded value is LOADx/2 | 0 |
| 8 | CCPxZAEN | Square wave mode enable bit for channel A (valid for PWM mode) 0: Disable 1: Enable, and duty cycle loaded value is LOADx/2 | 0 |
| 7 | - | Reserved | - |
| 6 | CCPxEN | CCPx enable bit 0: Disable 1: Enable | 0 |
| 5:4 | CCPxPS | CCPx prescale selection 0x0: PCLK 0x1: PCLK/4 0x2: PCLK/16 0x3: PCLK/64 | 0x0 |
| 3 | CCPxMS | CCPx mode selection 0: Capture mode 0 (valid when CAPEN=0) 1: PWM mode (valid when CAPEN=0) | 0 |
| 2 | CCPxCM0CS | CCPx capture mode 0 capture channel selection 0: Channel CCPxA 1: Channel CCPxB | 0 |
| 1:0 | CCPxCM0ES | CCPx capture mode 0 capture method selection 0x0: Start counting at CCPRUNx=1, capture on rising edge and generate an interrupt. 0x1: Start counting at CCPRUNx=1, capture on falling edge and generate an interrupt. 0x2: Start counting on the rising edge, capture on the falling edge and generate an interrupt. 0x3: Start counting on falling edge, capture on rising edge and generate an interrupt. | 0x0 |

12.5.2 CCP reload register (CCPLOADx) (x=0,1)

| Bit | Symbol | Description | Reset value |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:17 | - | Reserved | - |
| 16 | RELOAD | CCP0 module: PWM mode: Reload enable bit 0: Counter reload value is 0xFFFF 1: Counter reload value is CCP0LOAD Capture mode 0: 0: Counter reload value is 0xFFFF 1: Counter reload value is CCP0LOAD CCP1 module: PWM mode: Reload enable bit 0: Counter reload value is 0xFFFF 1: Counter reload value is CCP1LOAD Capture mode 0, 1: 0: Counter reload value is 0xFFFF 1: Counter reload value is CCP1LOAD | 0 |
| 15:0 | CCPxLOAD | CCPx counter reload value (It is recommended that the loaded value is not 0) | 0x0 |

12.5.3 CCPxA data register (CCPDxA) (x=0,1)

| Bit | Symbol | Description | Reset value |
|-------|-----------|---------------------------------------------------------------------------|-------------|
| 31:17 | - | Reserved | - |
| 16 | PWMxAOP | PWMxA output polarity selection 0: Normal output 1: Inverted output | 0 |
| 15:0 | CCPxADATA | PWM mode: PWMxA duty cycle Capture mode 0: Capture result | 0x0 |



12.5.4 CCPxB data register (CCPDxB) (x=0,1)

| Bit | Symbol | Description | Reset value |
|-------|-----------|---------------------------------------------------------------------------|-------------|
| 31:17 | - | Reserved | - |
| 16 | PWMxBOP | PWMxB output polarity selection 0: Normal output 1: Inverted output | 0 |
| 15:0 | CCPxBDATA | PWM mode: PWMxB duty cycle Capture mode: Capture result 0: | 0x0 |

12.5.5 CCP interrupt enable register (CCPIMSC)

| Bit | Symbol | Description | Reset value |
|-------|-----------|----------------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | CAP3IMSC | CAP3 capture interrupt enable bit 0: Disable 1: Enable | 0 |
| 10 | CAP2IMSC | CAP2 capture interrupt enable bit 0: Disable 1: Enable | 0 |
| 9 | CAP1IMSC | CAP1 capture interrupt enable bit 0: Disable 1: Enable | 0 |
| 8 | CAP0IMSC | CAP0 capture interrupt enable bit 0: Disable 1: Enable | 0 |
| 7:6 | - | Reserved | 0x0 |
| 5 | CCP1IMSC5 | CCP1 overflow interrupt enable bit 0: Disable 1: Enable | 0 |
| 4 | CCP0IMSC4 | CCP0 overflow interrupt enable bit 0: Disable 1: Enable | 0 |
| 3:2 | - | Reserved | - |
| 1 | CCP1IMSC1 | CCP1 compare/capture interrupt enable bit 0: Disable 1: Enable | 0 |
| 0 | CCP0IMSC0 | CCP0 compare/capture interrupt enable bit 0: Disable 1: Enable | 0 |



12.5.6 CCP interrupt source status register (CCPRIS)

| Bit | Symbol | Description | Reset value |
|-------|----------|-----------------------------------------------------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | CAP3RIS | CAP3 capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 10 | CAP2RIS | CAP2 capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 9 | CAP1RIS | CAP1 capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 8 | CAP0RIS | CAP0 capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 7:6 | - | Reserved | 0x0 |
| 5 | CCP1RIS5 | CCP1 overflow interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 4 | CCP0RIS4 | CCP0 overflow interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 3:2 | - | Reserved | - |
| 1 | CCP1RIS1 | CCP1 compare/capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |
| 0 | CCP0RIS0 | CCP0 compare/capture interrupt status bit 1: An interrupt is generated 0: No interrupt is generated | 0 |

12.5.7 CCP enabled interrupt status register (CCPMIS)

| Bit | Symbol | Description | Reset value |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | CAP3MIS | CAP3 enabled capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 10 | CAP2MIS | CAP2 enabled capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 9 | CAP1MIS | CAP1 enabled capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 8 | CAP0MIS | CAP0 enabled capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 7:6 | - | Reserved | - |
| 5 | CCP1MIS5 | CCP1 enabled overflow interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 4 | CCP0MIS4 | CCP0 enabled overflow interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 3:2 | - | Reserved | - |
| 1 | CCP1MIS1 | CCP1 enabled compare/capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |
| 0 | CCP0MIS0 | CCP0 enabled compare/capture interrupt status bit 1: Interrupt enable and an interrupt is generated 0: No interrupt is generated | 0 |

12.5.8 CCP interrupt clear register (CCPICLR)

| Bit | Symbol | Description | Reset value |
|-------|-----------|----------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | CAP3ICLR | Clear CAP3 capture interrupt status bit | 0 |
| 10 | CAP2ICLR | Write 1 to clear the CAP2 capture interrupt status bit | 0 |
| 9 | CAP1ICLR | Write 1 to clear the CAP1 capture interrupt status bit | 0 |
| 8 | CAP0ICLR | Write 1 to clear the CAP0 capture interrupt status bit | 0 |
| 7:6 | - | Reserved | - |
| 5 | CCP1ICLR5 | Write 1 to clear the CCP1 overflow interrupt status bit | 0 |
| 4 | CCP0ICLR4 | Write 1 to clear the CCP0 overflow interrupt status bit | 0 |
| 3:2 | - | Reserved | - |
| 1 | CCP1ICLR1 | Write 1 to clear the CCP1 compare/capture interrupt status bit | 0 |
| 0 | CCP0ICLR0 | Write 1 to clear the CCP0 compare/capture interrupt status bit | 0 |

12.5.9 CCP operation register (CCPRUN)

| Bit | Symbol | Description | Reset value |
|------|---------|-----------------------------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | CCPRUN1 | CCP1 operation control bit 0: Stop 1: Operate | 0 |
| 0 | CCPRUN0 | CCP0 operation control bit 0: Stop 1: Operate | 0 |

12.5.10 CCP write enable control register (LOCK)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0xaa, enable the registers with protection level P1A. When LOCK=0x55, enable the operation of registers with protection level P1B and P1A; when LOCK=other values, disable the operation of registers with protection level. | 0x0 |



12.5.11 CAP control register (CAPCON)

| Bit | Symbol | Description | Reset value |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:14 | - | Reserved | - |
| 13 | CAPEN2 | Capture mode 2 enable bit (only valid for CCP1) 0: -- 1: Capture Mode 2 enable bit, and disable Capture Mode 1 | 0 |
| 12 | CAPEN | Capture mode 1 enable bit 0: CCP0/CCP1 in PWM mode or Capture Mode 0 enabled 1: Capture Mode 1 enabled, i.e., full channel capture mode CCP0 can be set to Continuous Counting Mode CCP1 can be set to Continuous Counting Mode | 0 |
| 11 | CAP3RLEN | CAP3 capture in capture mode 1 triggers the CCP0 counter load enable bit 0: Disable 1: Enable, (Valid in Capture Mode 1 and CCP0 running state) When CAP3 captures a trigger signal, CCP0 will reload the data in the CCP0LOAD register during the counter's operation. | 0 |
| 10 | CAP2RLEN | CAP2 capture in capture mode 1 triggers the CCP0 counter load enable bit 0: Disable 1: Enable, (Valid in Capture Mode 1 and CCP0 running state) When CAP2 captures a trigger signal, CCP0 will reload the data in the CCP0LOAD register during the counter's operation. | 0 |
| 9 | CAP1RLEN | CAP1 capture in capture mode 1 triggers the CCP0 counter load enable bit 0: Disable 1: Enable, (Valid in Capture Mode 1 and CCP0 running state) When CAP1 captures a trigger signal, CCP0 will reload the data in the CCP0LOAD register during the counter's operation. | 0 |
| 8 | CAP0RLEN | CAP0 capture in capture mode 1 triggers the CCP0 counter load enable bit 0: Disable 1: Enable, (Valid in Capture Mode 1 and CCP0 running state) When CAP0 captures a trigger signal, CCP0 will reload the data in the CCP0LOAD register during the counter's operation. | 0 |
| 7:6 | CAP3ES | CAP3 capture mode selection 0x0: Disable 0x1: Rising edge capture 0x2: Falling edge capture 0x3: Both Edges | 0x0 |
| 5:4 | CAP2ES | CAP2 capture mode selection 0x0: Disable 0x1: Rising edge capture 0x2: Falling edge capture | 0x0 |

| | | | |
|-----|--------|-------------------------------------------------------------------------------------------------------------------------|-----|
| | | 0x3: Both Edges | |
| 3:2 | CAP1ES | CAP1 capture mode selection 0x0: Disable 0x1: Rising edge capture 0x2: Falling edge capture 0x3: Both Edges | 0x0 |
| 1:0 | CAP0ES | CAP0 capture mode selection 0x0: Disable 0x1: Rising edge capture 0x2: Falling edge capture 0x3: Both Edges | 0x0 |

12.5.12 CAP channel select register (CAPCHS)

| Bit | Symbol | Description | Reset value |
|-------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:17 | - | Reserved | - |
| 16 | ECAPS | ECAP capture channel group selection 0: Select ECAP00-ECAP02 1: Select ECAP10-ECAP13 | 0 |
| 15:12 | CAP3CHS | CAP3 capture channel selection 0x0: ECAPx0 (x=0 or 1, determined by ECAPS) 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: Disable 0x5: Disable 0x8: ACMP0 output (non-event output) 0x9: ACMP1 output (non-event output) 0xF: CCP1B Other value: Reserved | 0x0 |
| 11:8 | CAP2CHS | CAP2 capture channel selection 0x0: ECAPx0 (x=0 or 1, determined by ECAPS) 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: Disable 0x5: Disable 0xF: CCP1A Other value: Reserved | 0x0 |
| 7:4 | CAP1CHS | CAP1 capture channel selection 0x0: ECAPx0 (x=0 or 1, determined by ECAPS) 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: Disable 0x5: Disable 0xF: CCP0B Other value: Reserved | 0x0 |
| 3:0 | CAP0CHS | CAP0 capture channel selection 0x0: ECAPx0 (x=0 or 1, determined by ECAPS) 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: Disable 0x5: Disable 0xF: CCP0A Other value: Reserved | 0x0 |



12.5.13 CAP data register (CAP0DATA)

| Bit | Symbol | Description | Reset value |
|-------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | CAPXDATA | Read: Capture mode 2: After capturing, store CAPXDATA = Capture value of CAP1 - Capture value of CAP3. Other: - Write: Capture mode 2: - Other: 0x55aa, generate a capture operation for CAPn. Write: Other values, invalid. | 0x0 |
| 15:0 | CAP0DATA/ CAPYDATA | Read: Capture mode 2: After capturing, store CAPYDATA = Capture value of CAP1 - Capture value of CAP2. Other: Capture 16-bit value of CCP1 counter for CAP0. Write: Invalid | 0x0 |

12.5.14 CAP data register (CAPnDATA) (n=1-3)

| Bit | Symbol | Description | Reset value |
|-------|----------|---------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Read: Invalid Write: 0x55aa, generate a capture operation for CAPn. Write: Other values, invalid. | 0x0 |
| 15:0 | CAPnDATA | Read: Capture 16-bit value of CCP1 counter for CAPn. Write: Invalid | 0x0 |

Chapter 13 Enhanced PWM (EPWM)

13.1 Overview

The EPWM supports six PWM generators which can be configured as six independent PWM outputs, (EPWM0-EPWM5), or as three complementary PWM pairs (EPWM0-EPWM1, EPWM2-EPWM3, EPWM4-EPWM5) with three programmable dead-time generators.

Each pair of PWM shares an 8-bit prescaler and there are 6 sets of clock dividers available, providing 5 division factors (1, 1/2, 1/4, 1/8, 1/16). Each PWM output is controlled by an independent 16-bit counter, and a separate 16-bit comparator is used to adjust the duty cycle. The 6 PWM generators provide 28 interrupt flags, which are set when the period or duty cycle of the corresponding PWM channel matches the counter. Each PWM has its own enable bit.

Each PWM can be configured in one-shot mode (generating one PWM signal cycle) or continuous mode (continuously outputting PWM waveform).

13.2 Features

The enhanced PWM module has the following features:

- ◆ 6 independent 16-bit PWM control modes.
 - 6 independent outputs: EPWM0, EPWM1, EPWM2, EPWM3, EPWM4, EPWM5;
 - 3 sets of complementary PWM pairs: (EPWM0-EPWM1), (EPWM2-EPWM3), (EPWM4-EPWM5), with programmable dead-time insertion;
 - 3 sets of synchronized PWM pairs: (EPWM0-EPWM1), (EPWM2-EPWM3), (EPWM4-EPWM5), with synchronized PWM pair pins.
- ◆ Support group control, the outputs of EPWM0, EPWM2 and EPWM4 are synchronized, the outputs of PWM1, EPWM3 and EPWM5 are synchronized.
- ◆ One-shot mode (only supports edge-aligned) or auto-load mode.
- ◆ Support for edge-aligned and center-aligned modes.
- ◆ Center-aligned mode supports symmetric and asymmetric counting.
- ◆ Support programmable dead-time generators for complementary PWM.
- ◆ Each PWM has independent polarity control.
- ◆ Fault brake protection and recovery function (software/hardware-triggered and software/hardware recovery).
- ◆ ACMP analog comparator can trigger hardware brake protection.
- ◆ PWM edge or period can trigger AD conversion startup.

13.3 Function description

Explanation of related terms:

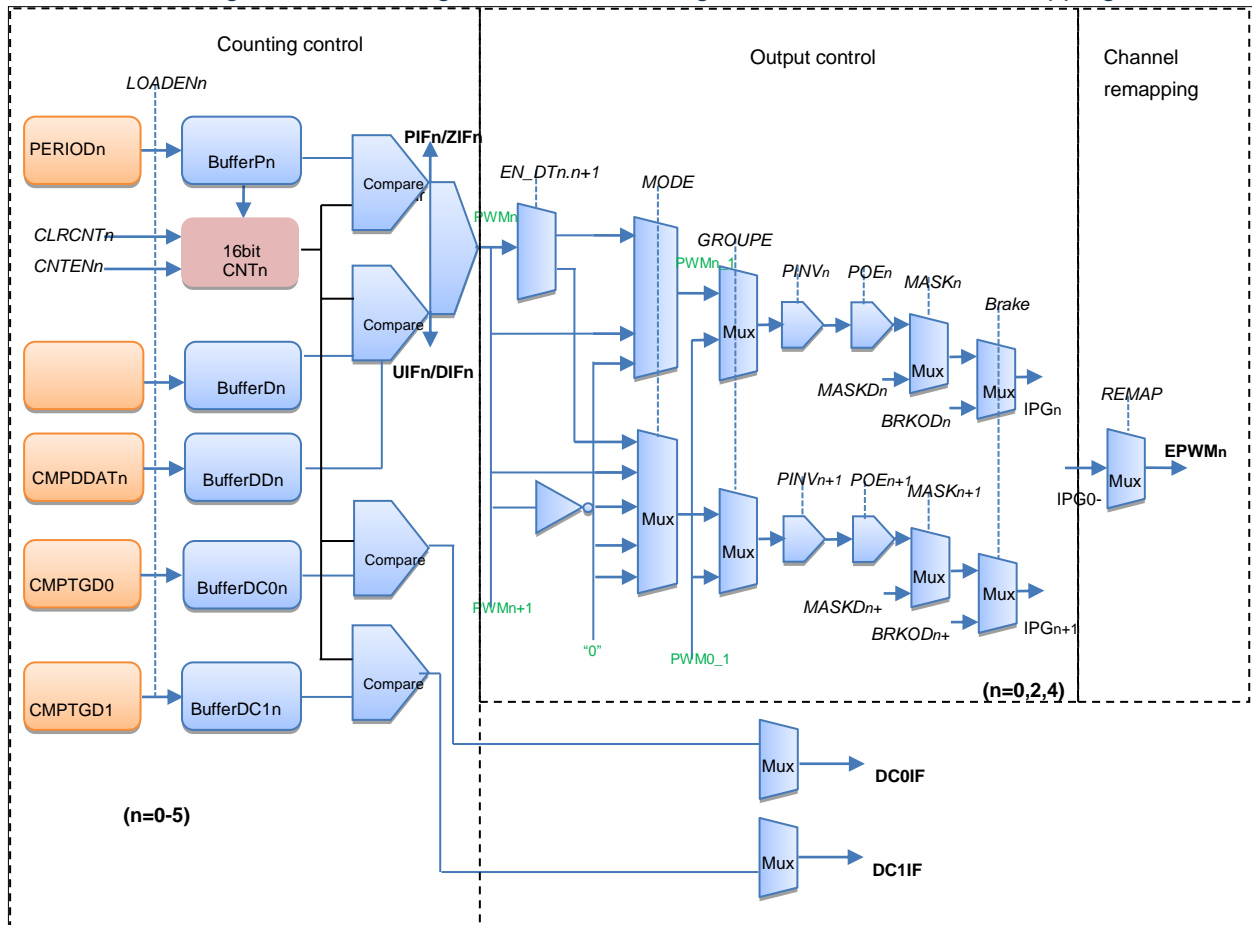
- 1) Period Point: When the counter CNTn counts to be equal to the period PERIODn, it is called the period point. The interrupt generated is PIFn.
- 2) Zero Point: When the counter CNTn counts to 0, it is called the zero point. The interrupt generated is ZIFn.
- 3) Up-Count Compare Point: When the counter CNTn incrementally counts to be equal to CMPDATn, it is called the up-count compare point. The interrupt generated is UIFn. There is no up-count compare point in edge-aligned counting mode.
- 4) Down-Count Compare Point: When the counter CNTn decrementally counts to be equal to CMPDATn or CMPDDATn, it is called the down-count compare point. The interrupt generated is DIFn.
- 5) Center Point: The center point refers to the moment when CNTn counts to be equal to CMPDATn or CMPDDATn in center-aligned counting mode. It is called the center point because CNTn will count down to 0 afterward, making it also a period point. There is no center point in edge-aligned counting mode, but there is a period point.

Note:

- 1) In edge-aligned mode, the period data is loaded at the start of the first count, which generates the period point. Since the counter counts to 0 afterward, the positions of subsequent period points coincide with zero points. This alignment mode has down-count compare points but no up-count compare points.
- 2) In center-aligned mode, the count starts from 0 and increments upward, generating the zero point. When it counts to the period data, it generates the period point (center point). The zero point alternates with the center point. This alignment mode has both up-count compare points and down-count compare points. In symmetric counting, both up-count compare points and down-count compare points are determined by CMPDATn. In asymmetric counting, the up-count compare point is determined by CMPDATn, and the down-count compare point is determined by CMDDATn.

13.3.1 Block diagram

Figure 13-1: The signal of IPG_n is the signal of EPWM_n before remapping



13.3.2 Clock division

Each PWM shares the same 8-bit prescaler, after the prescaler, each PWM can select 5 kinds (1, 1/2, 1/4, 1/8, 1/16) of prescaler ratios.

$PWM_CLK = PCLK / (CLKPSC_{xx} + 1) / CLKDIV_n$, here xx can be 01, 23, 45, $n=0-5$.

13.3.3 Independent output mode

The 6 EPWM channel outputs do not affect each other and operate according to their own period/duty cycle data.

13.3.4 Complementary output mode

In complementary output mode, the 6 PWM channels are divided into 3 pairs: 1 pair for EPWM0 and EPWM1, 1 pair for EPWM2 and EPWM3, and 1 pair for EPWM4 and EPWM5.

EPWM0-EPWM1 operate according to the period/duty cycle data of EPWM0, EPWM0 and EPWM1 waveforms are inverted.

EPWM2-EPWM3 operate according to the period/duty cycle data of EPWM2, EPWM2 and EPWM3 waveforms are inverted.

EPWM4-EPWM5 operate according to the period/duty cycle data of EPWM4, EPWM4 and EPWM5 waveforms are inverted.

In this mode, the EPWM1/EPWM3/EPWM5 outputs are independent of their associated runtime data registers, but the output controls remain active. For example, output enable, mask, brake, and so on.

Deadband delay control is supported in complementary mode.

13.3.5 Synchronous output mode

In synchronous output mode, the 6 PWMs are divided into 3 pairs, 1 pair for EPWM0 and EPWM1, 1 pair for EPWM2 and EPWM3, and 1 pair for EPWM4 and EPWM5.

EPWM0-EPWM1 operate according to the period/duty cycle data of EPWM0, EPWM0 and EPWM1 waveforms are in phase.

EPWM2-EPWM3 operate according to the period/duty cycle data of EPWM2, EPWM2 and EPWM3 waveforms are in phase.

EPWM4-EPWM5 operate according to the period/duty cycle data of EPWM4, EPWM4 and EPWM5 waveforms are in phase.

In this mode, EPWM1/EPWM3/EPWM5 outputs are independent of their own associated operation data registers, but the output control is still active. For example, output enable, mask, brake, and so on.

13.3.6 Group output mode

When GROUPEN=1 (enable grouping function), the 6 PWM channels are divided into 2 pairs: 1 pair for EPWM0, EPWM2, and EPWM4, and 1 pair for EPWM1, EPWM3, and EPWM5.

EPWM0-EPWM2-EPWM4 operate based on the period/duty cycle data of EPWM0, with all three channels having the same phase waveform.

EPWM1-EPWM3-EPWM5 operate based on the period/duty cycle data of EPWM1, with all three channels having the same phase waveform.

When the grouping function is enabled, the outputs of EPWM2, EPWM4, EPWM3, and EPWM5 are not dependent on their respective running data registers, but the output control (e.g., output enable, mask, brake control) still takes effect.

13.3.7 Load update mode

There are two types of counter loading modes: One-shot and Continuous (auto-loading mode).

One-shot mode:

Cycle duty cycle related data is loaded once at the beginning of the counter, and the output PWM cycle is related to the loading method.

LOADTYPn=0, edge alignment is 1 cycle, center alignment is 0.5 cycle.

LOADTYPn=1, edge alignment is 2 cycles, center alignment is 1 cycle.

LOADTYPn=2, edge alignment is 3 cycles, center alignment is 1.5 cycles.

LOADTYPn=3, edge alignment is 4 cycles, center alignment is 2 cycles.

Continuous mode:

In this mode, the duty cycle data is automatically loaded at zero point and center point within the PWM period. The center point loading only exists in center-aligned counting mode.

In edge-aligned counting mode, a zero point is generated along with a period point, and the counting comparison circuit reloads the values of CMPDATn/PERIODn/CMPTGD0/CMPTGD1.

In the center-aligned counting mode, both the center point and the zero point are automatically loaded with the values of the associated registers. This structure allows the first half of the waveform cycle duty cycle to be set differently from the second half of the waveform cycle duty cycle, and then remain the same when the period duty cycle related registers are not changed.

Due to the double-buffered structure of EPWM, when changing the values of the running registers such as CMPDATn/CMPDDATn/PERIODn/CMPTGD0/CMPTGD1 during EPWM operation, the PWM output waveform will not change immediately. Only at the zero-point or period point, these register values will be loaded into the corresponding buffer.

With this structure, after changing the duty cycle data, the current PWM period or half-period output waveform will not immediately change. The PWM waveform will only change in the next period or half-period. That means any changes to PWM-related data will not affect a current complete PWM period or half-period.

In high-speed applications, it is possible that the loading point has arrived, but the write operation to the running registers has not been completed. In this case, it is not expected to have partial running data loaded while the other part is not.

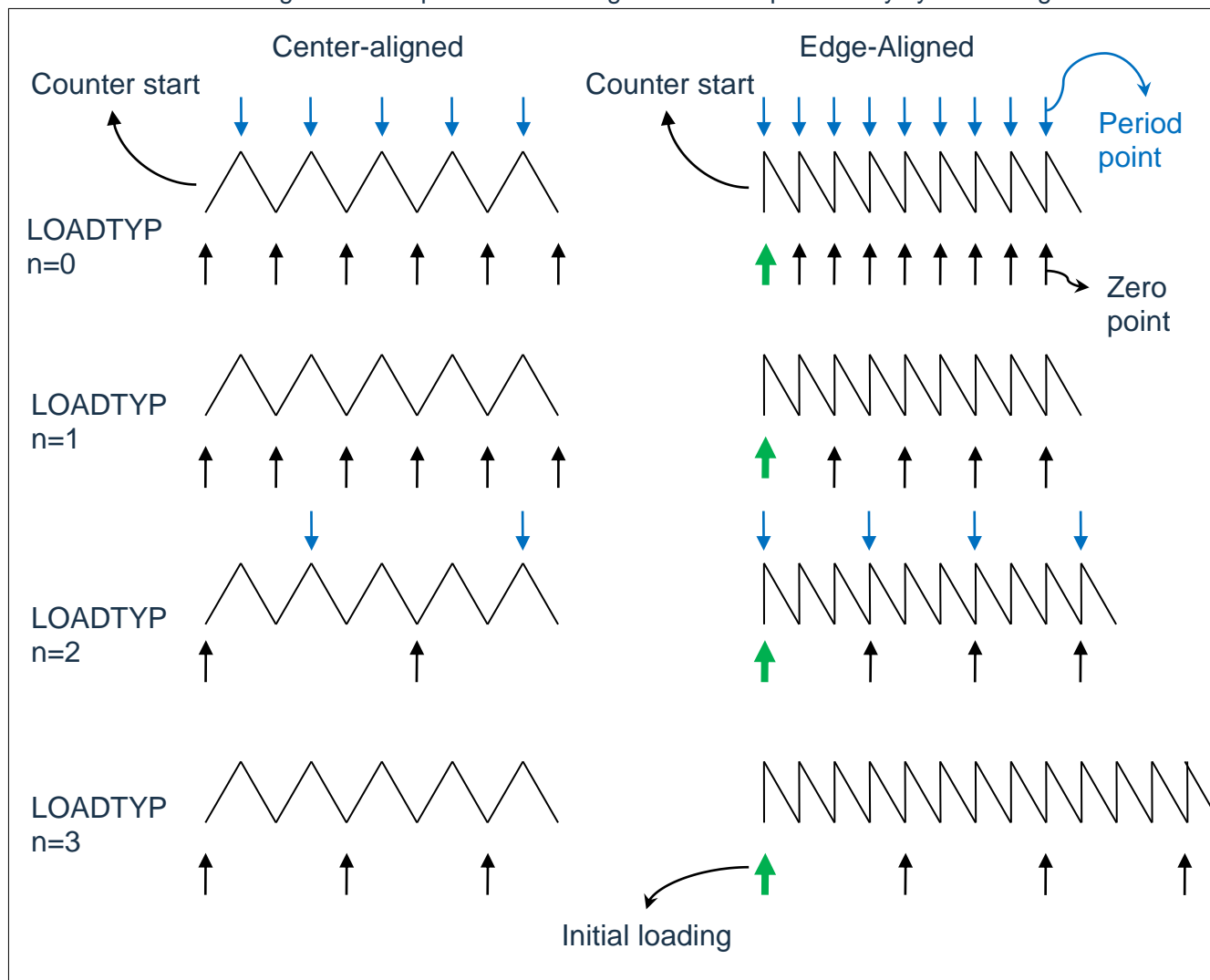
To address this high-speed application scenario, the EPWM module provides a loading enable bit. After changing the relevant running registers, the loading enable bit LOADENn needs to be set to 1. After the loading is completed, the LOADENn bit will automatically be cleared. Additionally, the state of this bit can be read to determine whether the values of the relevant registers have been loaded into the actual circuit. If LOADENn=0, it means that the values have been loaded and will affect the output PWM waveform. If LOADENn=1, it means that the values have not been loaded yet and the current PWM waveform has not changed. The values of the previously changed registers will be loaded at the next loading point. If the relevant running registers are changed again, the LOADENn bit also needs to be set to 1 again.

By default, PWM will load the running data of the relevant registers at both zero point and period point and generate zero point and period point interrupts. To adapt to more flexible application requirements, PWM supports different loading methods and zero point/period point interrupt generation methods.

In the register EPWMCON3, LOADTYPn (0-5) can be set to determine the loading method and the interrupt generation method for the zero point/period point:

| LOADTYEn | Center-aligned loading | Edge-aligned loading |
|----------|--------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| 00 | Load and generate zero point and period point interrupt flags at each zero point or period point | Load and generate zero point and period point interrupt flags at each zero point or period point |
| 01 | Load and generate zero point interrupt flags at each zero point | Load and generate zero point interrupt flags at every 2 zero points |
| 10 | Alternate load and generate zero point and period point interrupt flags between the first zero point and the next period point | Load and generate zero-point and period point interrupt flags at every 3 zero points or period points |
| 11 | Load and generate related zero point interrupt flags every two zero points | Load and generate zero point interrupt flags at every 4 zero points |

Figure 13-2: Updated block diagram of PWM period/duty cycle loading



13.3.8 Edge-aligned counting mode

In edge-aligned mode, with counting down method, the 16-bit PWM counter CNTn starts counting down at the beginning of each cycle. It compares with the latched value CMPDATn, and when CNTn=CMPDATn, EPWMn outputs a high-level signal and sets CMPnDIF to 1. The CNTn continues counting down until it reaches 0, at which point EPWMn outputs a low-level signal. When PWMnCNTM=1, the current CMPDATn and PERIODn will be reloaded, and PIF (period interrupt flag) will be set.

Edge-aligned related parameters:

High level time=(CMPDATn+1) ×Tpwm

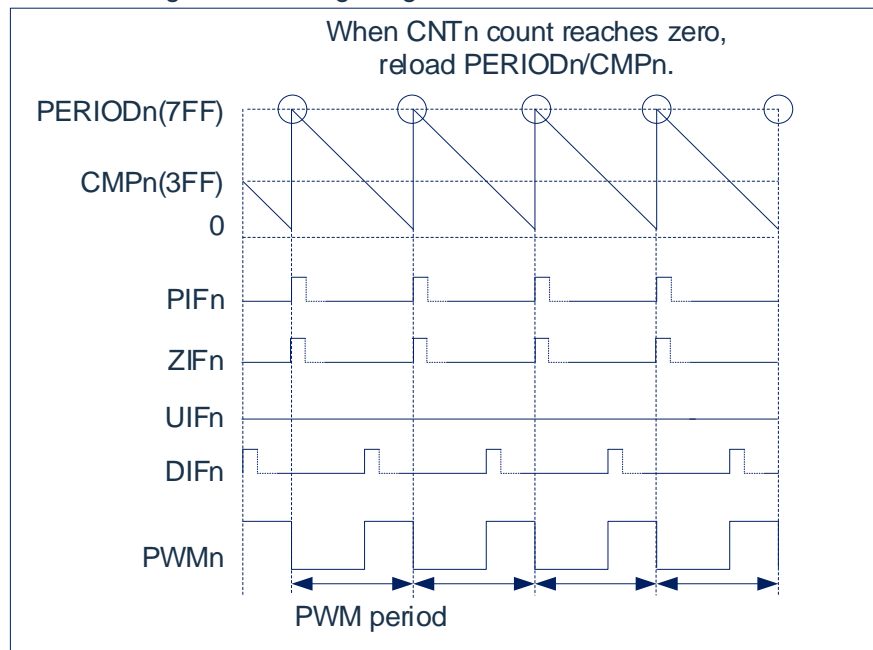
Period=(PERIODn+1) ×Tpwm

Duty cycle= $\frac{\text{CMPDATn}+1}{\text{periodn}+1}$

If CMPDATn > PERIODn, the duty cycle is 100%, and the EPWMn channel remains high. It will not generate a down-compare interrupt.

If CMPDATn = 0, the duty cycle is 0%.

Figure 13-3: Edge alignment mode waveform



13.3.9 Center-aligned counting mode

In center-aligned mode, the counting process starts by counting up and then counts down.

Center-aligned mode can be further divided into two types: symmetric counting mode and asymmetric counting mode.

In symmetric counting mode (ASYMEN=0), the duty cycle is determined by CMPDATAn.

In asymmetric counting mode (ASYMEN=1), the duty cycle is determined by both CMPDATAn and CMPDDATn.

In center-aligned symmetric counting mode, the 16-bit PWM counter CNTn starts counting up from 0. When CNTn reaches CMPDATn, EPWMn outputs a high level. Then, CNTn continues counting up until it reaches the value of PERIODn. After that, CNTn starts counting down. During the counting down process, when CNTn = CMPDATn, EPWMn outputs a low level. Then, it continues counting down until it reaches 0.

High level time=(PERIODn×2-CMPDATn×2-1)×Tpwm

Period=(PERIODn)×2×Tpwm

Duty cycle= $\frac{\text{PERIODn} \times 2 - \text{CMPDATn} \times 2 - 1}{\text{periodn} \times 2}$

If CMPDATn is greater than or equal to PERIODn, the duty cycle is 0%, and EPWMn channel remains low, without generating any up-compare or down-compare interrupts.

If PERIODn is 0, the duty cycle is 0%, and EPWMn channel remains low, with zero interrupts and period-point interrupts present as long as CNTn is enabled.

If CMPDATn is 0, the duty cycle is 100%.

Figure 13-4: Center-aligned mode symmetric counting waveform

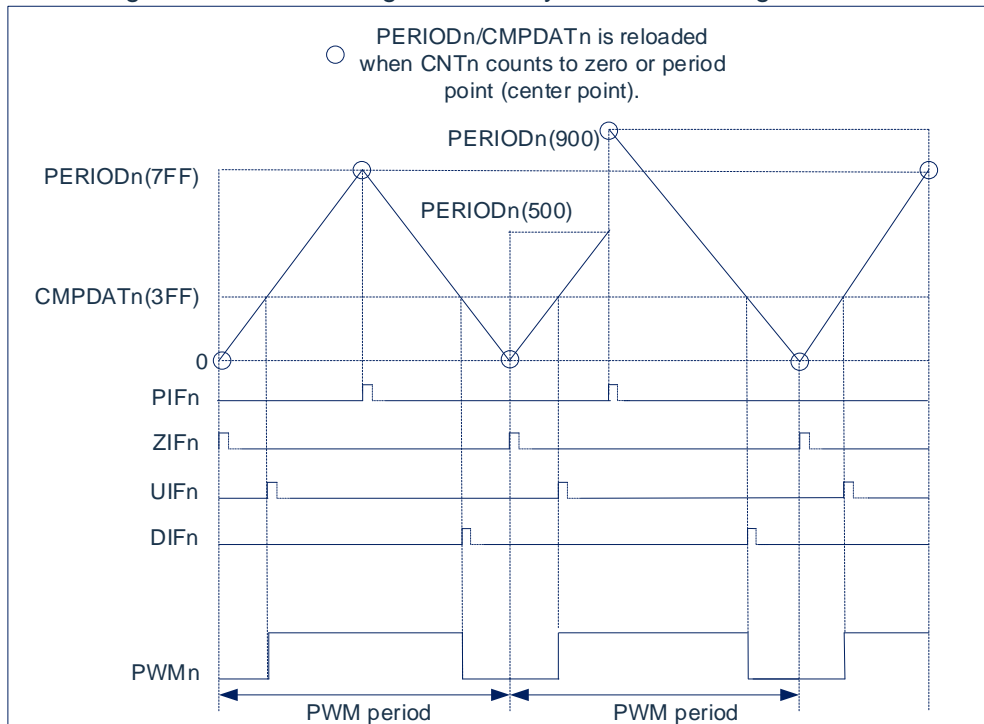
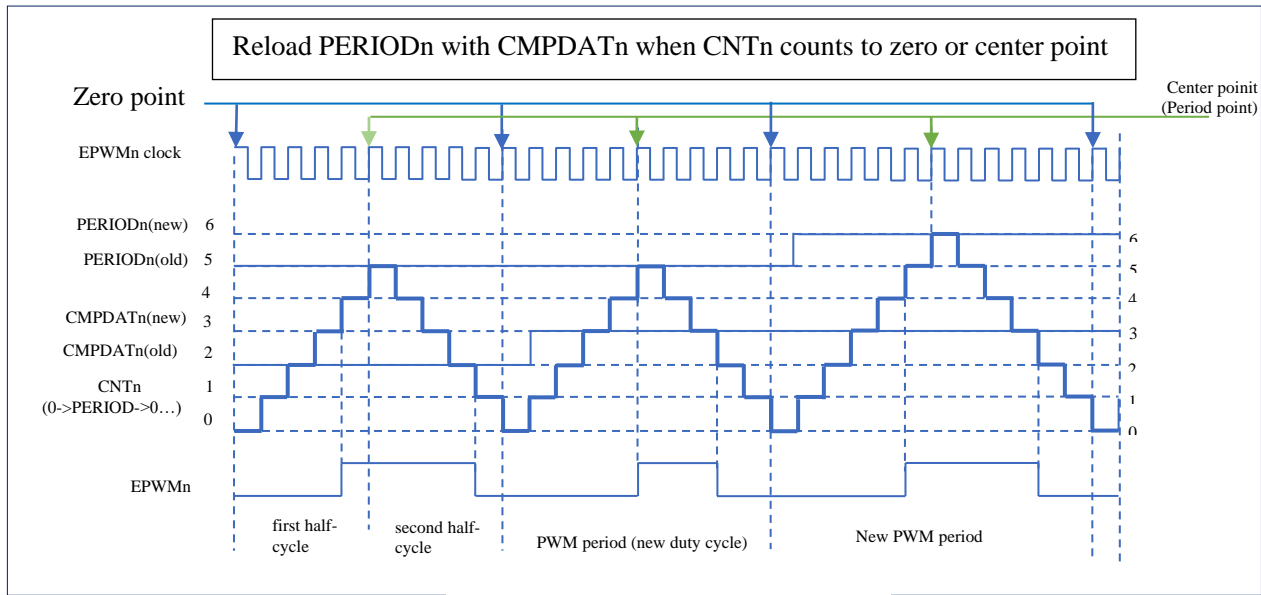


Figure 13-5: Center-aligned counter waveform (symmetric counting)



In center-aligned asymmetric counting mode, the 16-bit PWM counter CNTn starts counting from 0 and increments upward. When CNTn = CMPDATn, EPWMn outputs a high level. After that, CNTn continues counting upward until it reaches PERIODn. Then, CNTn starts counting downward. During the downward counting process, when CNTn = CMPDDATn, EPWMn outputs a low level. Afterward, it continues counting downward until it reaches 0. To enable center-aligned asymmetric counting mode, ASYMEN needs to be set to 1. The asymmetric counting mode can realize the precise center-aligned waveform.

The parameters related to center-aligned asymmetric counting mode are as follows:

High level time = $(\text{PERIODn} \times 2 - \text{CMPDDATn} - \text{CMPDATn}) \times T_{\text{pwm}}$

Duty cycle = $\left(\frac{\text{PERIODn} \times 2 - \text{CMPDDATn} - \text{CMPDATn} - 1}{\text{PERIODn} \times 2} \right)$, ($\text{CMPDATn} < \text{PERIODn}$, $\text{CMPDDATn} < \text{PERIODn}$)

Duty cycle = $\left(\frac{\text{PERIODn} - \text{CMPDDATn} - \text{CMPDATn} - 1}{\text{PERIODn} \times 2} \right)$, ($\text{CMPDATn} \geq \text{PERIODn}$, $\text{CMPDDATn} < \text{PERIODn}$)

Duty cycle = $\left(\frac{\text{PERIODn} - \text{CMPDDATn}}{\text{PERIODn} \times 2} \right)$, ($\text{CMPDATn} < \text{PERIODn}$, $\text{CMPDDATn} \geq \text{PERIODn}$)

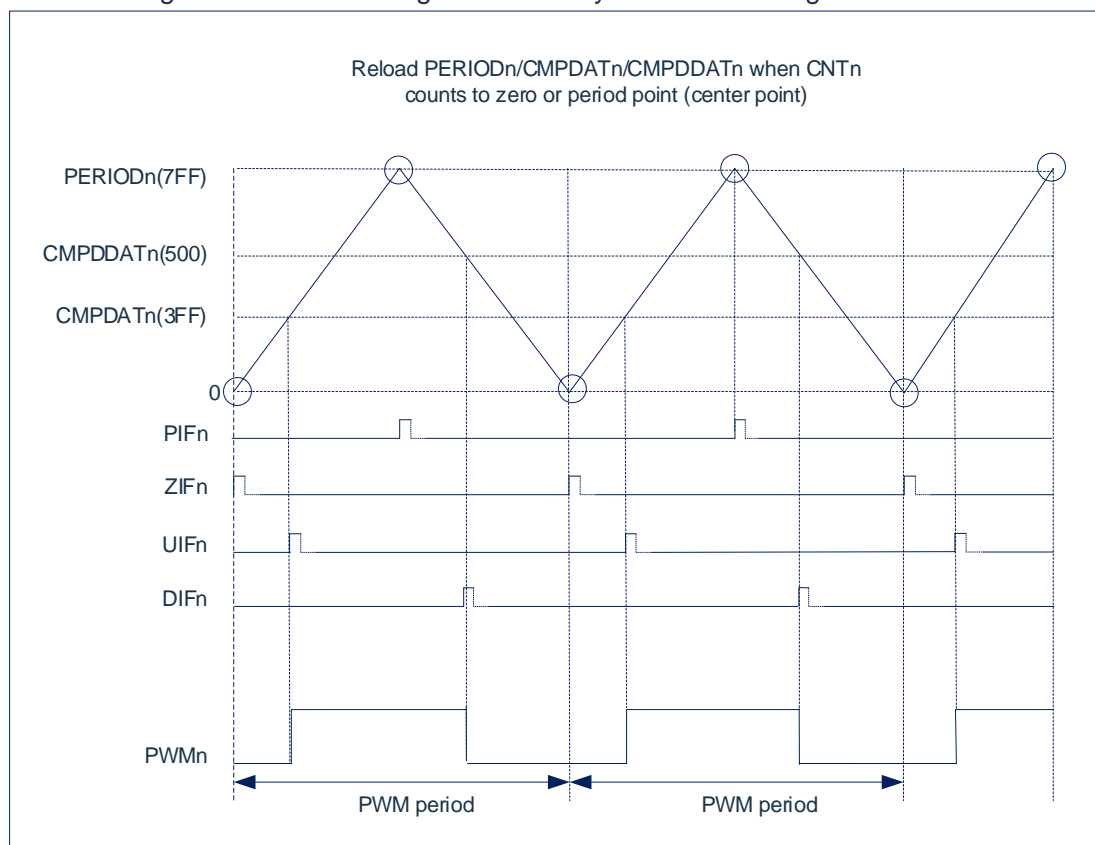
Duty cycle = 0%, ($\text{CMPDATn} \geq \text{PERIODn}$, $\text{CMPDDATn} < \text{PERIODn}$)

CMPDATn ≥ PERIODn does No up-compare interrupt is generated when CMPDATn ≥ PERIODn.
No down-compare interrupt is generated when CMPDATn ≥ PERIODn.

If PERIODn = 0, the duty cycle is 0%, the EPWMn channel is always low, and the zero interrupt and period point interrupt always exist when CNTn is enabled.

If CMPDATn = 0 and CMDATDn = 0, the duty cycle is 100%.

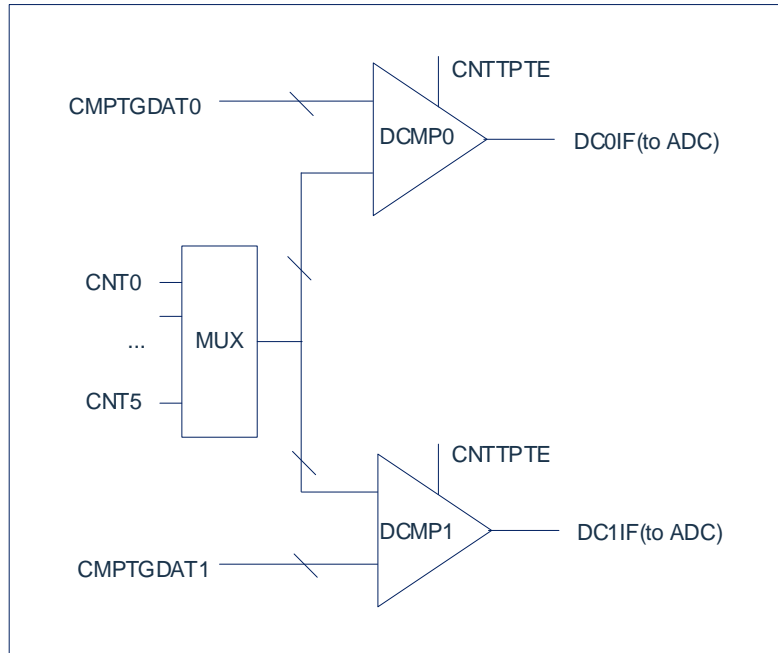
Figure 13-6: Center-aligned mode asymmetric counting waveform



13.3.10 Independent counter compare function

During the counting of the PWM_n channel counter (CNT_n), two digital comparators are provided to compare the counter value with pre-set values. If the counter value equals the pre-set value, an interrupt signal or ADC trigger can be generated. This function does not affect the PWM output.

Figure 13-7: Independent counter compare function

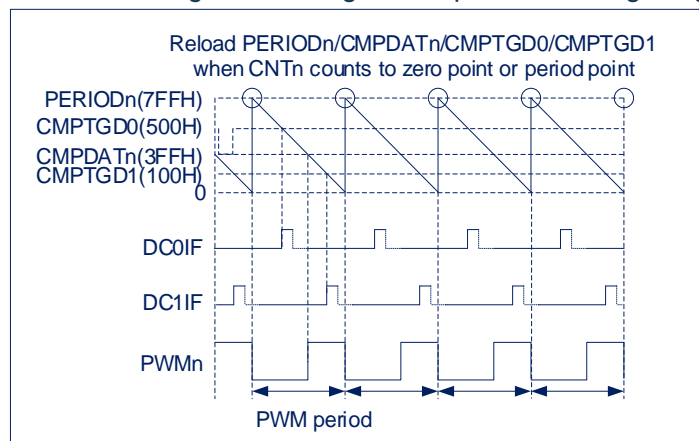


Digital Comparator 0 compares the value of CNT_n with CMPTGDAT0. If they are equal, the interrupt flag DC0IF is generated. CMPTGD0[10:8] selects one of the PWM0-5 channel counters to compare with CMPTGDAT0.

Digital Comparator 1 compares the value of CNT_n with CMPTGDAT1. If they are equal, the interrupt flag DC1IF is generated. CMPTGD1[10:8] selects one of the PWM0-5 channel counters to compare with CMPTGDAT1.

- 1) In edge-aligned mode, the working mode of digital comparators is as follows:

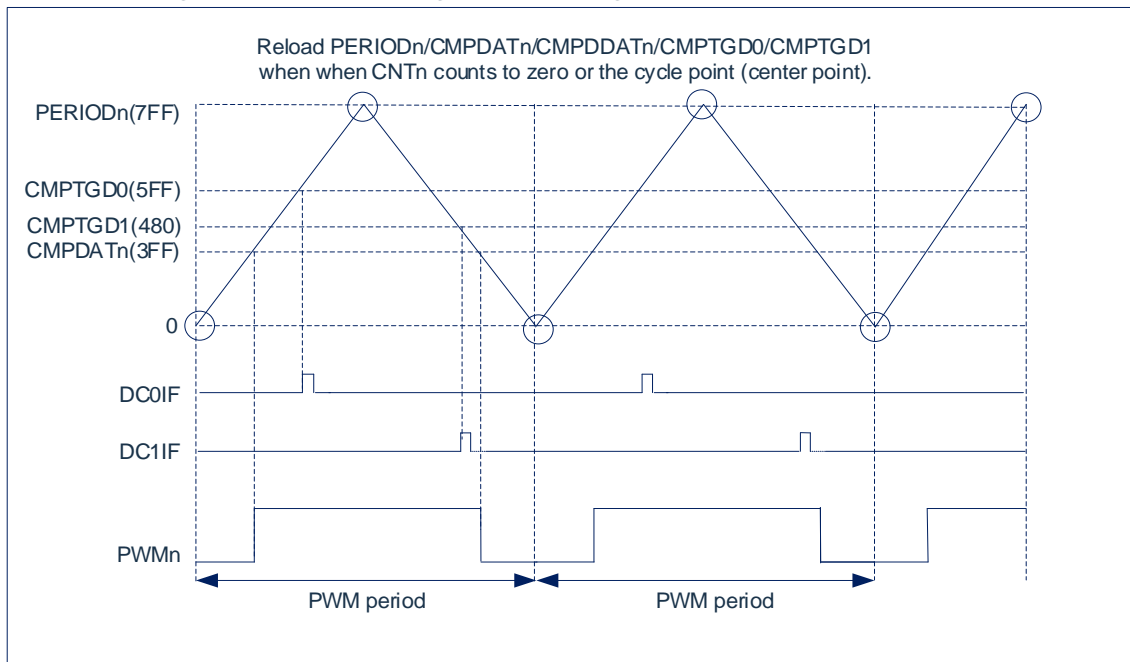
Figure 13-8: Working mode of digital comparators in edge-aligned mode



In edge counting mode, digital comparator 0/1 can be set to generate a compare interrupt at any counting moment.

2) Center-aligned mode, digital comparator operation method:

Figure 13-9: Center-aligned mode, digital comparator operation method



In center-aligned counting mode, digital comparators 0/1 can each be set to trigger in either upward or downward counting mode. That is, both can be triggered in the first half-cycle or the second half-cycle, or one can be triggered in the first half-cycle and the other in the second half-cycle. This is determined by the $CMPTGD0[19]$ bit $CMPTGDSn$.

13.3.11 Programmable dead-time generator

The 6-channel PWM can be configured into 3 complementary pairs. In the complementary output mode, the period and duty cycle of PWM1, PWM3, and PWM5 are determined by the corresponding registers of PWM0, PWM2, and PWM4, respectively. At the same time, the dead-time delay register can also affect the duty cycle of the complementary PWM pairs. In this mode, apart from the corresponding output enable control bit (PWMnOE), the output waveforms of PWM1/PWM3/PWM5 are no longer controlled by their own registers.

In the complementary mode, each complementary PWM pair supports the insertion of dead-time delay. The inserted dead-time is calculated as follows:

PWM0/1 dead-time: $(\text{PWM01DT}[9:0]+1) \cdot \text{TPWM0}$

PWM2/3 dead-time: $(\text{PWM23DT}[9:0]+1) \cdot \text{TPWM2}$

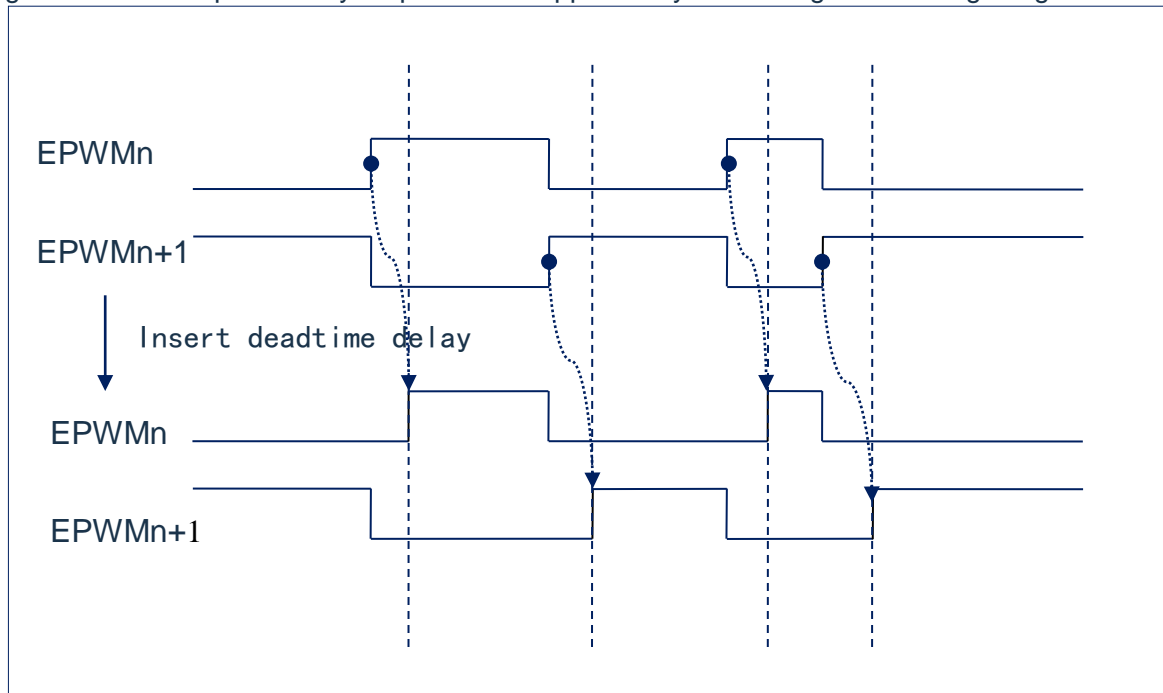
PWM4/5 dead-time: $(\text{PWM45DT}[9:0]+1) \cdot \text{TPWM4}$

TPWM0/TPWM2/TPWM4 represent the clock source periods of PWM0/PWM2/PWM4, respectively.

The range of dead-time can be set from 0.021us to 21us ($F_{\text{pwmn}}=48\text{MHz}$).

The output mode does not affect the counter mode, so both center-aligned and edge-aligned modes support the complementary output mode.

Figure 13-10: Complementary output mode supported by center-aligned and edge-aligned modes.



13.3.12 Mask and mask preset function

EPWM supports the mask function. Each channel of EPWM0-EPWM5 has individual control, and the corresponding control bits for EPWMn are MASKENn and MASKDn (in the MASK register).

When MASKENn=0, the EPWMn channel outputs the normal PWM waveform.

When MASKENn=1, the EPWMn channel outputs the data from MASKDn.

The control register MASK for the mask function also supports the automatic loading of preset values. To enable this feature, set the MASKLE bit in the output control register POEN to 1, allowing MASK to automatically load the value from the MASKNXT register, while disabling writing to the MASK register.

The loading time is determined by the MASKLS<2:0> bits in POEN, which can be set to match the load cycle/duty (loading point) of one of EPWM0-EPWM5.

13.3.13 Hall sensor interface function

EPWM considers the interface with a Hall sensor. It includes an internal HALL position detection circuit that detects the levels of the filtered CCP0/1 module's internal capture channels CAP0, CAP1, and CAP2.

After internal processing, the detection circuit produces a state called HALLST:

HALLST has eight states, corresponding to the HALL position states as follows:

| HALLST | Corresponding state |
|--------|----------------------------------------------------------------------|
| 000 | HALL detection circuit not started or initial state |
| 001 | {CAP2-CAP0}=001 |
| 010 | {CAP2-CAP0}=010 |
| 011 | {CAP2-CAP0}=011 |
| 100 | {CAP2-CAP0}=100 |
| 101 | {CAP2-CAP0}=101 |
| 110 | {CAP2-CAP0}=110 |
| 111 | Error state during {CAP2-CAP0} change process or incorrect sequence. |

The value of HALLST can be read from the MASKNXT register, allowing the HALL position or sequence state to be determined at any time.

The HALL state detection sequence supports the following two orders ({CAP2, CAP1, CAP0} appearing in a sequence):

- ◆ -6-2-3-1-5-4-6-
- ◆ -6-4-5-1-3-2-6-

If any other sequence occurs, it is considered an error, and HALLST will enter the 111 state and stop the detection. It will also generate an interrupt flag HALLIF. To restart the HALL detection circuit, set the HALLCLR bit in the MASKNXT register to 1, and HALLST will shift from the 111 state to the initial 000 state to restart the detection circuit.

The HALL detection circuit provides a feature that can automatically load related functions with the mask. This feature allows controlling the output channel waveform of EPWM without software intervention.

Each valid state of HALLST corresponds to a mask preset cache, and there are totally seven mask preset caches:

| HALLST(HALLEN=1) | Corresponding mask preset cache: |
|------------------|----------------------------------|
| 000 | Mask preset cache 7 |
| 001 | Mask preset cache 1 |
| 010 | Mask preset cache 2 |
| 011 | Mask preset cache 3 |
| 100 | Mask preset cache 4 |
| 101 | Mask preset cache 5 |
| 110 | Mask preset cache 6 |
| 111 | Mask preset cache 7 |
| HALLEN=0 | Mask preset cache 0 |

If the feature of automatic loading of mask preset values is enabled, then at the corresponding state and at the selected loading point, the data in the corresponding mask preset cache will be loaded into the MASK register. For example:

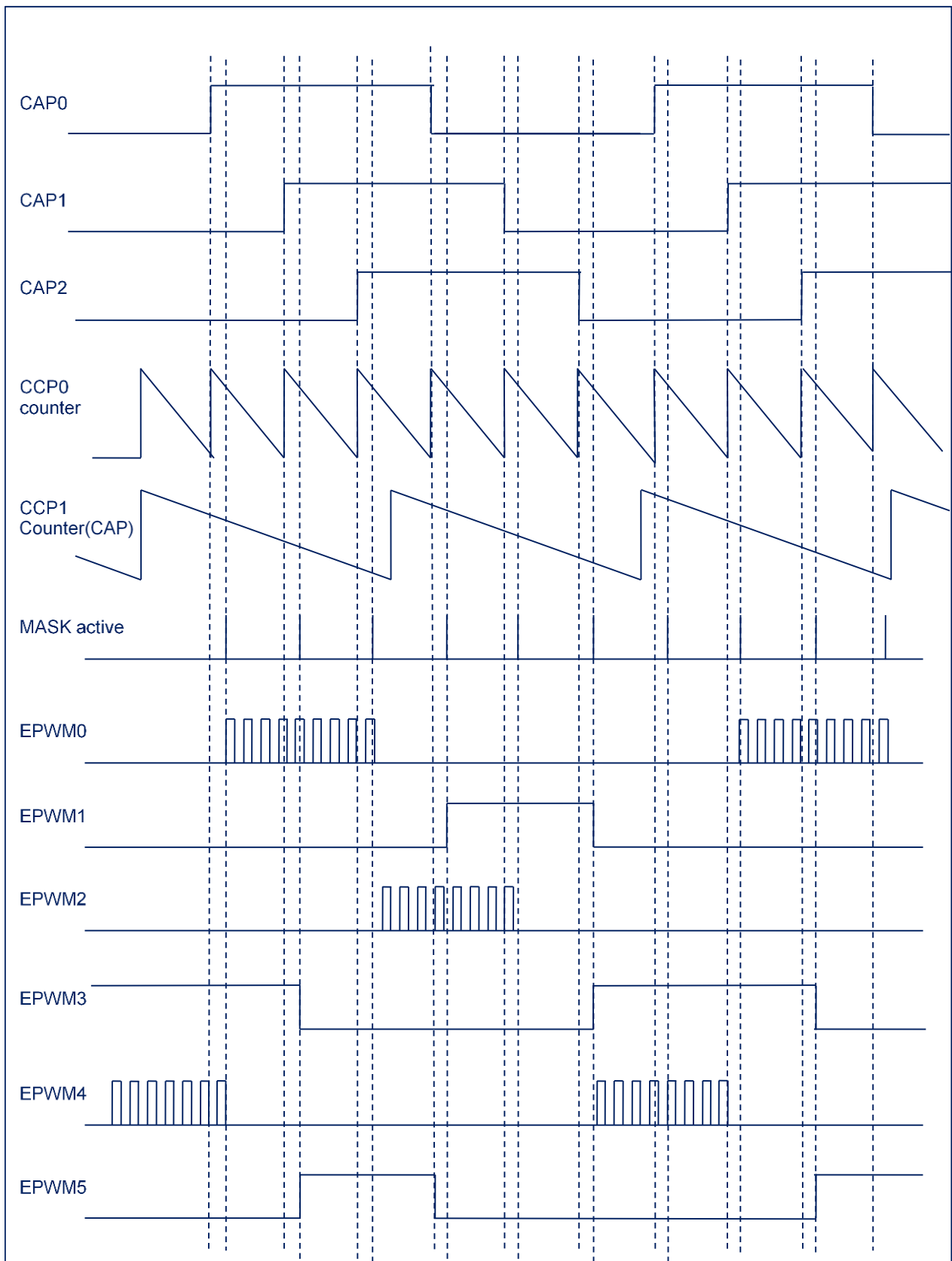
When the position state in HALLST changes from 000 to 001 and enters the first loading point of state 001, the data in Mask Preset Cache 1 will be loaded into the MASK register.

Later, when the position state in HALLST changes from 001 to 101 and enters the first loading point of state 101, the data in Mask Preset Cache 5 will be loaded into the MASK register.

If an incorrect sequence occurs, such as the CAP2-CAP0 input changing from 101 to 010, which is not the correct sequence, the position state in HALLST will change from 101 to 111, and the interrupt flag HALLIF will be set to 1. At the first loading point of state 111, the data in Mask Preset Cache 7 will be loaded into the MASK register.

In the initial state, the data in Mask Preset Cache 7 is loaded into the MASK register at the loading point.

Figure 13-11: Example of HALL detection timing (this does not represent the actual running waveform)



13.3.14 Fault protection function (brake and recovery function)

EPWM supports fault protection function, and BKODn controls the brake threshold for 6 channels. The fault protection function is controlled by the BRKCTL register.

The triggering sources for EPWM fault protection are as follows:

Level-triggered sources:

- 1) External BKIN level signal (high or low level)
- 2) Software brake signal (SWBRK set to 1)
- 3) Analog comparator 0 output (high or low output)
- 4) Analog comparator 1 output (high or low output) pulse-triggered sources:
- 5) Edge signal of external BKIN (rising edge or falling edge)
- 6) Analog comparator 0 output event (rising edge, falling edge, or both edges)
- 7) Analog comparator 1 output event (rising edge, falling edge, or both edges)
- 8) ADC result comparator 0 event (result compare event)

Fault interrupt flag bit BRKIF (Cleared to 0 by software):

After detecting a valid brake trigger source signal, the fault interrupt flag BRKIF is set to 1 and needs to be cleared to 0 by software.

Fault signal flag bit BRKAF (read-only):

The fault signal flag BRKAF is set to 1, and it automatically clears to 0 when the brake signal is revoked. BRKAF is a read-only bit.

Fault protection output status flag bit BRKOSF (read-only):

When BRKOSF is 1, it indicates that EPWMn channel outputs the BRKODn data state;

When BRKOSF is 0, it indicates that EPWMn is in normal output state.

It indicates whether the EPWM output is in brake state or normal state. BRKOSF will be set to 1 when a valid brake signal is detected. In software recovery mode, a brake clear operation (BRKCLR=1) will affect the state of this bit.

Fault protection modes can be divided into 4 types to meet different requirements in fault protection scenarios.

| BRKMS | Fault protection mode |
|-------|-------------------------------------------|
| 00 | Stop Mode (Software Recovery) |
| 01 | Pause Mode (Software Recovery) |
| 10 | Recovery Mode (Hardware Recovery) |
| 11 | Delayed Recovery Mode (Hardware Recovery) |

Note: The fault interrupt flag (BRKIF) is unrelated to the recovery function and only represents the occurrence of a brake signal. The fault interrupt flag also supports accumulation function.

Stop Mode:

Generate fault protection and fault interrupt flags, clear the CNTENn bit to 0, and stop the counter operation. To recover the output, the brake signal needs to be revoked, and the fault state clearing operation (BRKCLR=1) needs to be executed, then set CNTENn to 1 again.

Pause Mode:

Generate fault protection and fault interrupt flags, but the counter continues to operate. To recover the output, revoke the brake signal, execute the fault state clearing operation (BRKCLR=1), and restore normal output at the most recent load update point.

Recovery mode:

Generate fault protection and fault interrupt flags, but the counter continues to operate. After revoking the brake signal, the normal output automatically restores at the most recent load update point. There is no need to execute the fault state clearing operation.

Pay attention to distinguish whether the brake signal is a pulse signal or a level signal: If the brake source is a level signal, the output can only be restored after the brake is revoked; if it is a pulse signal, the EPWM output restores at the most recent load update point after triggering the brake, unless another brake pulse signal is generated during this period.

Delayed recovery mode:

Generate fault protection and fault interrupt flags, but the counter continues to operate. After revoking the brake signal, the EPWM restores normal output after a delay time at the most recent load update point. There is no need to execute the fault state clearing operation.

The delay time can be freely set, and the low 16-bit RDT of BRKRDT control the delay time. The delay time is as follows:

$$T_{delay} = RDT * TAPBCLK$$

Pay attention to distinguish whether the brake signal is a pulse signal or a level signal: If the brake source is a level signal, the output can only be restored after the brake is revoked; if it is a pulse signal, the EPWM output waits for the completion of the delay time and then restores at the most recent load update point, unless another brake pulse signal is generated during this period.

After generating the brake protection, EPWMn channel outputs the data in BRKODn. Each channel can independently set the output to high or low level.

13.3.15 Output status in debug mode

In debug mode, the CPU has two states: operation state and pause state. The operation state is the normal execution state, while the pause state occurs after executing a STOP instruction, reaching a breakpoint, or stepping.

In the pause state, the output status of EPWMn (with POEn=1) can be configured using the HALTMS bit in the CON register.

When HALTMS=0, the output status of EPWMn remains normal during the pause.

When HALTMS=1, the output status of EPWMn during the pause becomes the brake data, but no fault-related flags are generated. The EPWMn counter continues to run, and the EPWMn output is restored to the nearest load update point when the running state is resumed.

It should be noted that in debug mode, the values of the relevant operational data registers of EPWMn do not automatically change and will retain their previous states.

13.3.16 Output channel remapping

The output channel remapping function allows for more flexible arrangement of channels in applications. By default, the pins corresponding to EPWM0-EPWM5 in the chip pinout diagram are assigned to their respective PWM channels. However, the output channel remapping function can be used to reconfigure the desired channels.

The internal channels corresponding to EPWM0-EPWM5 are IPG0-IPG5 by default. Through the POREMAP register, any one of the IPG0-IPG5 channels can be reassigned to EPWMn (n=0-5). The output channel remapping function only affects the remapping of port output channels, while their internal control and interrupts remain unchanged.

13.3.17 EPWM configuration process

- ◆ Enable EPWM register operation by writing 0x55 to the LOCK register.
- ◆ Configure EPWM clock division and set the pre-divider ratio and independent divider ratio.
- ◆ Select the mode, either independent mode or complementary mode.
- ◆ Set the EPWM period and duty cycle.
- ◆ Set the EPWM output polarity.
- ◆ Enable the EPWM counter.
- ◆ Configure the relevant IO ports as EPWM functionality.
- ◆ Enable the output of the corresponding EPWM channels.
- ◆ Write 0x00 to the LOCK register to prevent accidental operation on EPWM-related registers until the next operation on EPWM-related registers is needed, at which point re-enable it.

13.3.18 Interrupts

The EPWM unit has eight interrupt sources:

- ZIFn – An interrupt flag generated when the EPWM counter counts to zero.
- UIFn - An interrupt flag generated when the EPWM counter counts up to CMPDATn.
- PIFn - An interrupt flag generated during edge-aligned counting or center-aligned counting of the EPWM counter.
- DIFn - An interrupt flag generated when the EPWM counter counts down to CMPDATn/CMPDDATn.
- DC0IF - An interrupt flag generated when the EPWM counter counts to the value equal to CMPTGD0.
- DC1IF - An interrupt flag generated when the EPWM counter counts to the value equal to CMPTGD1.
- HALLIF - Hall state error interrupt flag.
- BRKIF - Fault interrupt flag.

All interrupt flags are set by hardware and must be cleared by software.

13.4 Register mapping

(EPWM base address = 0x4006_4200)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|--------------------------|--------------|-----|-----------------------------------------------|-------------|
| CLKPSC _(P1B) | 0x000 | R/W | EPWM Prescaler Register | 0x0 |
| CLKDIV _(P1B) | 0x004 | R/W | EPWM Clock Select Register | 0x0 |
| CON _(P1B) | 0x008 | R/W | EPWM Control Register | 0x0 |
| CON2 _(P1B) | 0x00C | R/W | EPWM Control Register 2 | 0x0 |
| CON3 _(P1B) | 0x010 | R/W | EPWM Control Register 3 | 0x0 |
| PERIOD0 _(P1A) | 0x014 | R/W | EPWM Period Register 0 | 0x0 |
| PERIOD1 _(P1A) | 0x018 | R/W | EPWM Period Register 1 | 0x0 |
| PERIOD2 _(P1A) | 0x01C | R/W | EPWM Period Register 2 | 0x0 |
| PERIOD3 _(P1A) | 0x020 | R/W | EPWM Period Register 3 | 0x0 |
| PERIOD4 _(P1A) | 0x024 | R/W | EPWM Period Register 4 | 0x0 |
| PERIOD5 _(P1A) | 0x028 | R/W | EPWM Period Register 5 | 0x0 |
| CMPDAT0 _(P1A) | 0x02C | R/W | EPWM Compare Register 0 | 0x0 |
| CMPDAT1 _(P1A) | 0x030 | R/W | EPWM Compare Register 1 | 0x0 |
| CMPDAT2 _(P1A) | 0x034 | R/W | EPWM Compare Register 2 | 0x0 |
| CMPDAT3 _(P1A) | 0x038 | R/W | EPWM Compare Register 3 | 0x0 |
| CMPDAT4 _(P1A) | 0x03C | R/W | EPWM Compare Register 4 | 0x0 |
| CMPDAT5 _(P1A) | 0x040 | R/W | EPWM Compare Register 5 | 0x0 |
| POREMAP _(P1B) | 0x044 | R/W | EPWM Output Channel Remapping Register | 0x543210 |
| POEN _(P1B) | 0x048 | R/W | EPWM Output Control Register | 0x0 |
| BRKCTL _(P1B) | 0x04C | R/W | EPWM Fault Protection Control Register | 0x0 |
| DTCTL _(P1B) | 0x050 | R/W | EPWM Dead Time Control Register | 0x0 |
| MASK _(P1B) | 0x054 | R/W | EPWM Output Mask Register | 0x0 |
| MASKNXT _(P1B) | 0x058 | R/W | EPWM Output Mask Preset Register | 0x0 |
| CMPTGD0 _(P1B) | 0x05c | R/W | EPWM Counter Compare Register 0 | 0x0 |
| CMPTGD1 _(P1B) | 0x060 | R/W | EPWM Counter Compare Register 1 | 0x0 |
| IMSC _(P1B) | 0x064 | R/W | EPWM Interrupt Enable Register | 0x0 |
| RIS | 0x068 | RO | EPWM Interrupt Source Status Register | 0x0 |
| MIS | 0x06c | RO | EPWM Enabled Interrupt Status Register | 0x0 |
| ICLR | 0x070 | WO | EPWM Interrupt Clear Register | 0x0 |
| IFA _(P1B) | 0x074 | R/W | EPWM Interrupt Accumulation Control Register | 0x0 |
| LOCK | 0x078 | R/W | EPWM Write Enable Control Register | 0x0 |
| BRKRDT _(P1B) | 0x07C | R/W | EPWM Fault Protection Recovery Delay Register | 0x0 |

Note:

The registers marked with (P1A/P1B) are protected registers.

(P1A): When LOCK==55H or AAH, the marked registers allow writing; when it equals any other value, writing is prohibited.

(P1B): When LOCK==55H, the marked registers allow writing; when it equals any other value, writing is prohibited.

13.5 Register description

13.5.1 EPWM prescaler register (CLKPSC)

| Bit | Symbol | Description | Reset value |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:24 | - | Reserved | - |
| 23:16 | CLKPSC45 | EPWM counter 4 and 5 clock prescaler $CLK_PSC45 = PCLK / (CLKPSC45 + 1)$ If CLKPSC45=0, the pre-scaler has no clock output. If the CLKDIVn bit selects a clock related to PSC, the counter does not operate. | 0x0 |
| 15:8 | CLKPSC23 | EPWM counter 2 and 3 clock prescaler $CLK_PSC23 = PCLK / (CLKPSC23 + 1)$ If CLKPSC23=0, the pre-scaler has no clock output. If the CLKDIVn bit selects a clock related to PSC, the counter does not operate. | 0x0 |
| 7:0 | CLKPSC01 | EPWM counter 0 and 1 clock prescaler $CLK_PSC01 = PCLK / (CLKPSC01 + 1)$ If CLKPSC01=0, the pre-scaler has no clock output. If the CLKDIVn bit selects a clock related to PSC, the counter does not operate. | 0x0 |

13.5.2 EPWM clock selection register (CLKDIV)

| Bit | Symbol | Description | Reset value |
|-------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:23 | - | Reserved | - |
| 22:20 | CLKDIV5 | Counter 5 clock division frequency selection 000: CLK_PSC45/2 001: CLK_PSC45/4 010: CLK_PSC45/8 011: CLK_PSC45/16 100: CLK_PSC45/1 Other value: PCLK | 0x0 |
| 19 | - | Reserved | - |
| 18:16 | CLKDIV4 | Counter 4 clock division frequency selection 000: CLK_PSC45/2 001: CLK_PSC45/4 010: CLK_PSC45/8 011: CLK_PSC45/16 100: CLK_PSC45/1 Other value: PCLK | 0x0 |
| 15 | - | Reserved | - |
| 14:12 | CLKDIV3 | Counter 3 clock division frequency selection 000: CLK_PSC23/2 001: CLK_PSC23/4 010: CLK_PSC23/8 011: CLK_PSC23/16 100: CLK_PSC23/1 Other value: PCLK | 0x0 |
| 11 | - | Reserved | - |
| 10:8 | CLKDIV2 | Counter 2 clock division frequency selection 000: CLK_PSC23/2 001: CLK_PSC23/4 010: CLK_PSC23/8 011: CLK_PSC23/16 100: CLK_PSC23/1 Other value: PCLK | 0x0 |
| 7 | - | Reserved | - |
| 6:4 | CLKDIV1 | Counter 1 clock division frequency selection 000: CLK_PSC01/2 001: CLK_PSC01/4 010: CLK_PSC01/8 011: CLK_PSC01/16 100: CLK_PSC01/1 Other value: PCLK | 0x0 |
| 3 | - | Reserved | - |
| 2:0 | CLKDIV0 | Counter 0 clock division frequency selection 000: CLK_PSC01/2 001: CLK_PSC01/4 010: CLK_PSC01/8 | 0x0 |

| | | | |
|--|--|------------------------------------------------------------|--|
| | | 011: CLK_PSC01/16 100: CLK_PSC01/1 Other value: PCLK | |
|--|--|------------------------------------------------------------|--|

13.5.3 EPWM control register (CON)

| Bit | Symbol | Description | Reset value |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:27 | - | Reserved | - |
| 26 | HALTMS | EPWMn channel status control bit during HALT (debug pause) (If POENn=0, the output of EPWMn is in high resistance state) 0: All channels output normally (POENn=1) 1: All channels output brake data (POENn=1). (In the debug state, the output of EPWMn is the brake data after running to a breakpoint/single step or pausing after operating the STOP button.) | 0 |
| 25:24 | MODE | EPWM operating mode selection 00: Independent mode 01: Complementary mode 10: Synchronous mode 11: Reserved | 0x0 |
| 23 | GROUNPEN | EPWM grouping enable bit 0: All PWM channels are independent of each other 1: EPWM0 control EPWM2, EPWM4, EPWM1 control EPWM3, EPWM5 | 0 |
| 22 | ASYMEN | Asymmetric count enable in EPWM center alignment mode 0: Symmetric count enable 1: Asymmetric count enable | 0 |
| 21 | CNTTYPE | EPWM count alignment selection 0: Edge alignment 1: Center alignment | 0 |
| 20:19 | - | Reserved | - |
| 18 | EN_DT45 | EPWM counter 4 and 5 deadband enable bit 0: Disable counter 4 and 5 deadband 1: Enable counter 4 and 5 deadband | 0 |
| 17 | EN_DT23 | EPWM counter 2 and 3 deadband enable bit 0: Disable counter 2 and 3 deadband 1: Enable counter 2 and 3 deadband | 0 |
| 16 | EN_DT01 | EPWM counter 0 and 1 deadband enable bit 0: Disable counter 0 and 1 deadband 1: Enable counter 0 and 1 deadband | 0 |
| 15:14 | - | Reserved | - |
| 13 | PINV5 | EPWM5 output polarity control bit 0: Normal output 1: Inverted output | 0 |
| 12 | PINV4 | EPWM4 output polarity control bit 0: Normal output 1: Inverted output | 0 |

| | | | |
|-----|----------|-----------------------------------------------------------------------------|---|
| 11 | PINV3 | EPWM3 output polarity control bit 0: Normal output 1: Inverted output | 0 |
| 10 | PINV2 | EPWM2 output polarity control bit 0: Normal output 1: Inverted output | 0 |
| 9 | PINV1 | EPWM1 output polarity control bit 0: Normal output 1: Inverted output | 0 |
| 8 | PINV0 | EPWM0 output polarity control bit 0: Normal output 1: Inverted output | 0 |
| 7:6 | - | Reserved | - |
| 5 | CNTMODE5 | EPWM5 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |
| 4 | CNTMODE4 | EPWM4 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |
| 3 | CNTMODE3 | EPWM3 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |
| 2 | CNTMODE2 | EPWM2 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |
| 1 | CNTMODE1 | EPWM1 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |
| 0 | CNTMODE0 | EPWM0 auto-load/one-shot mode 0: One-shot mode 1: Auto-load mode | 0 |

13.5.4 EPWM control register (CON2)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:6 | - | Reserved | - |
| 5 | CNTEN5 | EPWM5 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |
| 4 | CNTEN4 | EPWM4 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |
| 3 | CNTEN3 | EPWM3 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |
| 2 | CNTEN2 | EPWM2 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |
| 1 | CNTEN1 | EPWM1 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |
| 0 | CNTEN0 | EPWM0 counter enable bit 0: Disable 1: Enable (The bit is cleared automatically after one-shot mode completion) | 0 |

13.5.5 EPWM control register (CON3)

| Bit | Symbol | Description | Reset value |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | LOADNWINT | <p>EPWM load interrupt flag-related control bit</p> <p>1: Load control is not related to the interrupt flag</p> <p>0: Load control is related to the interrupt flag</p> <p>When the load action is generated, whether the interrupt flag is generated with the load or not, if not, the interrupt flag will be generated at every period and zero point.</p> | 0 |
| 30 | LETGHALL | <p>HALL status trigger LOADENn enable bit</p> <p>0: Disable</p> <p>1: Enable HALL state change trigger LOADENn=1</p> <p>Note: If the HALL detection status changes, set the load enable bits of EPWM0-EPWM5 to 1.</p> | 0 |
| 29 | LETGACMP1 | <p>ACMP1 trigger LOADENn enable bit</p> <p>0: Disable</p> <p>1: Trigger LOADENn=1 when enabling ACMP1</p> <p>Note: If an ACMP1 event is generated, set the load enable bits of EPWM0-EPWM5 to 1.</p> | 0 |
| 28 | LETGACMP0 | <p>ACMP0 trigger LOADENn enable bit</p> <p>0: Disable</p> <p>1: Trigger LOADENn=1 when enabling ACMP0</p> <p>Note: If an ACMP0 event is generated, set the load enable bits of EPWM0-EPWM5 to 1.</p> | 0 |
| 27:26 | LOADTYP5 | <p>EPWM5 load/interrupt mode selection bit</p> <p>00: Interrupt flags are loaded and generated at each zero and period point.</p> <p>01: Each zero point is loaded with a generated interrupt flag</p> <p>10: The first zero point is loaded alternately with the next period point with the generation of interrupt flags</p> <p>11: Load & generate interrupt flags every two zero points</p> | 0x0 |
| 25:24 | LOADTYP4 | <p>EPWM4 load/interrupt mode selection bit</p> <p>00: Interrupt flags are loaded and generated at each zero and period point.</p> <p>01: Each zero point is loaded with a generated interrupt flag</p> <p>10: The first zero point is loaded alternately with the next period point with the generation of interrupt flags</p> <p>11: Load & generate interrupt flags every two zero points</p> | 0x0 |

| | | | |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 23:22 | LOADTYP3 | EPWM3 load/interrupt mode selection bit 00: Interrupt flags are loaded and generated at each zero and period point. 01: Each zero point is loaded with a generated interrupt flag The first zero point is loaded alternately with the next period point with the generation of interrupt flags 10: 11: Load & generate interrupt flags every two zero points | 0x0 |
| 21:20 | LOADTYP2 | EPWM2 load/interrupt mode selection bit 00: Interrupt flags are loaded and generated at each zero and period point. 01: Each zero point is loaded with a generated interrupt flag 10: The first zero point is loaded alternately with the next period point with the generation of interrupt flags 11: Load & generate interrupt flags every two zero points | 0x0 |
| 19:18 | LOADTYP1 | EPWM1 load/interrupt mode selection bit 00: Interrupt flags are loaded and generated at each zero and period point. 01: Each zero point is loaded with a generated interrupt flag 10: The first zero point is loaded alternately with the next period point with the generation of interrupt flags 11: Load & generate interrupt flags every two zero points | 0x0 |
| 17:16 | LOADTYP0 | EPWM0 load/interrupt mode selection bit 00: Interrupt flags are loaded and generated at each zero and period point. 01: Each zero point is loaded with a generated interrupt flag 10: The first zero point is loaded alternately with the next period point with the generation of interrupt flags 11: Load & generate interrupt flags every two zero points | 0x0 |
| 15:14 | - | Reserved | - |
| 13 | LOADEN5 | EPWM5 period/comparator load enable 0: Disable 1: Enable (Automatically cleared by hardware after loading) | 0 |
| 12 | LOADEN4 | EPWM4 period/comparator load enable 0: Disable 1: Enable (Automatically cleared | 0 |

| | | by hardware after loading) | |
|-----|---------|------------------------------------------------------------------------------------------------------------------|---|
| 11 | LOADEN3 | EPWM3 period/comparator load enable 0: Disable 1: Enable (Automatically cleared by hardware after loading) | 0 |
| 10 | LOADEN2 | EPWM2 period/comparator load enable 0: Disable 1: Enable (Automatically cleared by hardware after loading) | 0 |
| 9 | LOADEN1 | EPWM1 period/comparator load enable 0: Disable 1: Enable (Automatically cleared by hardware after loading) | 0 |
| 8 | LOADEN0 | EPWM0 period/comparator load enable 0: Disable 1: Enable (Automatically cleared by hardware after loading) | 0 |
| 7:6 | - | Reserved | - |
| 5 | CNTCLR5 | EPWM5 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |
| 4 | CNTCLR4 | EPWM4 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |
| 3 | CNTCLR3 | EPWM3 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |
| 2 | CNTCLR2 | EPWM2 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |
| 1 | CNTCLR1 | EPWM1 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |
| 0 | CNTCLR0 | EPWM0 counter clear bit 0: Disable 1: Enable (Automatically cleared by hardware) | 0 |

13.5.6 EPWM period register 0-5 (PERIOD0-5)

| Bit | Symbol | Description | Reset value |
|-------|---------|----------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:0 | PERIODn | EPWMn counter period value | 0x0 |

13.5.7 EPWM compare register 0-5(CMPDAT0-5)

| Bit | Symbol | Description | Reset value |
|-------|----------|----------------------------------|-------------|
| 31:16 | CMPDDATn | EPWMn counter down compare value | 0x0 |
| 15:0 | CMPDATn | EPWMn counter compare value | 0x0 |

13.5.8 EPWM output control register (POEN)

| Bit | Symbol | Description | Reset value |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | MASKLE | EPWM mask control preset data load enable bit 0: Disable 1: Enable (Enable MASKNXT register to load data into MASK register, and disable writing to MASK register. In addition, when this bit is set to 1, the mask data is not loaded immediately, but only when the corresponding load point is reached.) | 0 |
| 10:8 | MASKLS | EPWM mask control data load time select bit 000: Load at the EPWM0 load point 001: Load at the EPWM1 load point 010: Load at the EPWM2 load point 011: Load at the EPWM3 load point 100: Load at the EPWM4 load point 101: Load at the EPWM5 load point 11x: Reserved | 0x0 |
| 7:6 | - | Reserved | - |
| 5:0 | POENn | EPWMn output enable bit 0: EPWM channel n output disable 1: EPWM channel n output enable | 0x0 |

13.5.9 EPWM output channel remap register (POREMAP)

| Bit | Symbol | Description | Reset value |
|-------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:24 | PWMRMEN | EPWM channel remap function enable control AAH: Remap function enable Selection of EPWMn channel output by PWMnRM Other: Remap function disable The EPWMn fixed channel outputs are as follows: EPWM0<- IPG0 EPWM1<- IPG1 EPWM2<- IPG2 EPWM3<- IPG3 EPWM4<- IPG4 EPWM5<- IPG5 | 0x0 |
| 23 | - | Reserved | - |
| 22:20 | PWM5RM | EPWM channel 5 remap select bit 000: Map the output of IPG0 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | 0x5 |
| 19 | - | Reserved | - |
| 18:16 | PWM4RM | EPWM channel 4 remap select bit 000: Map the output of IPG0 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | 0x4 |
| 15 | - | Reserved | - |
| 14:12 | PWM3RM | EPWM channel 3 remap select bit 000: Map the output of IPG0 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | 0x3 |
| 11 | - | Reserved | - |
| 10:8 | PWM2RM | EPWM channel 2 remap select bit 000: Map the output of IPG0 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | 0x2 |
| 7 | - | Reserved | - |
| 6:4 | PWM1RM | EPWM channel 1 remap select bit 000: Map the output of IPG0 | 0x1 |

| | | | |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | |
| 3 | - | Reserved | - |
| 2:0 | PWM0RM | EPWM channel 0 remap select bit 000: Map the output of IPG0 001: Map the output of IPG1 010: Map the output of IPG2 011: Map the output of IPG3 100: Map the output of IPG4 101: Map the output of IPG5 11x: Reserved | 0x0 |

13.5.10 EPWM fault protection control register (BRKCTL)

| Bit | Symbol | Description | Reset value |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | BRKEN | EPWM fault protection function general enable bit 0: Disable (reset fault protection circuit) 1: Enable | 0 |
| 30 | BRKAF | EPWM fault signal flag bit (read-only) 0: No fault generated 1: Fault signal generated or brake signal remains valid | 0 |
| 29:28 | BRKMS | EPWM fault protection mode selection bit 00: Stop mode 01: Pause mode 10: Recovery mode 11: Delayed recovery mode Note: When switching to the fault protection mode, the fault protection enable must be disabled first, then switch to the fault protection mode, and finally, enable the fault protection enable bit. | 0x0 |
| 27 | BRKCLR | EPWM fault protection clear bit (write-only) 0: -- 1: Clear the fault protection status Note: only when BRKAF=0 can write 1 to perform fault clear operation, otherwise the operation is invalid. | 0 |
| 26:24 | BRKRCS | EPWM fault recovery load point selection bit 000: EPWM0 load point recovery 001: EPWM1 load point recovery 010: EPWM2 load point recovery 011: EPWM3 load point recovery 100: EPWM4 load point recovery 101: EPWM5 load point recovery Other: Disable selection | 0x0 |
| 23 | ACMP1BKLE | Analog comparator 1 output level control brake enable bit 0: Disable 1: Enable | 0 |
| 22 | ACMP1BKLS | Analog comparator 1 output level control brake selection bit 0: Low level generates brake 1: High level generates brake | 0 |
| 21 | ACMP0BKLE | Analog comparator 0 output level control brake enable bit 0: Disable 1: Enable | 0 |
| 20 | ACMP0BKLS | Analog comparator 0 output level control brake selection bit | 0 |

| | | | |
|----|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| | | 0: Low level generates brake 1: High level generates brake | |
| 19 | ACMP1BKEN | Analog comparator 1 output event control brake enable bit 0: Disable 1: Enable (Comparator output event refers to generating rising edge/falling edge/double edge, which can be selected in ACMP->CEVCON) | 0 |
| 18 | ACMP0BKEN | Analog comparator 0 output event control brake enable bit 0: Disable 1: Enable (Comparator output event refers to generating rising edge/falling edge/double edge, which can be selected in ACMP->CEVCON) | 0 |
| 17 | ADCMP1BKEN | ADC comparator 1 output brake enable bit 0: Disable 1: Enable | 0 |
| 16 | ADCMP0BKEN | ADC comparator 0 output brake enable bit 0: Disable 1: Enable | 0 |
| 15 | -- | Must be set to 0 | 0 |
| 14 | -- | Reserved | 0 |
| 13 | BRKOSF | EPWM fault protection output status flag bit (read-only) 0: EPWMn channel is in normal output state 1: EPWMn channel is in output BRKODn data state | 0 |
| 12 | SWBRK | Software brake enable bit 0: Disable software brake 1: Immediately generate software brake | 0 |
| 11 | EXTBRKEE | External hardware brake edge detection enable bit 0: Disable 1: Enable | 0 |
| 10 | EXTBRKES | External hardware brake edge detection selection bit 0: Falling edge triggers brake 1: Rising edge triggers brake | 0 |
| 9 | EXTBRKLE | External hardware brake level detection enable bit 0: Disable 1: Enable | 0 |
| 8 | EXTBRKLS | External hardware brake level detection selection bit 0: Low level generates brake | 0 |

| | | | |
|-----|--------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | 1: High level generates brake | |
| 7:6 | - | Must be set to 0 | 0x0 |
| 5:0 | BRKODn | EPWMn brake output level selection bit 0: After fault brake, channel n outputs low level 1: After fault brake, channel n outputs high level | 0x0 |

13.5.11 EPWM dead time control register (DTCTL)

| Bit | Symbol | Description | Reset value |
|-------|--------|-----------------------------------------------------------------------------|-------------|
| 31:30 | - | Reserved | - |
| 29:20 | DTI45 | Channel 4 and 5 dead time control register Dead time = PWM_CLK45 × DTI45 | 0x0 |
| 19:10 | DTI23 | Channel 2 and 3 dead time control register Dead time = PWM_CLK23 × DTI23 | 0x0 |
| 9:0 | DTI01 | Channel 0 and 1 dead time control register Dead time = PWM_CLK01 × DTI01 | 0x0 |

13.5.12 EPWM mask output control register (MASK)

| Bit | Symbol | Description | Reset value |
|-------|---------|---------------------------------------------------------|-------------|
| 31:14 | - | Reserved | - |
| 13 | MASKEN5 | EPWM5 mask output enable bit 0: Disable 1: Enable | 0 |
| 12 | MASKEN4 | EPWM4 mask output enable bit 0: Disable 1: Enable | 0 |
| 11 | MASKEN3 | EPWM3 mask output enable bit 0: Disable 1: Enable | 0 |
| 10 | MASKEN2 | EPWM2 mask output enable bit 0: Disable 1: Enable | 0 |
| 9 | MASKEN1 | EPWM1 mask output enable bit 0: Disable 1: Enable | 0 |
| 8 | MASKEN0 | EPWM0 mask output enable bit 0: Disable 1: Enable | 0 |
| 7:6 | - | Reserved | - |
| 5 | MASKD5 | EPWM5 mask data 0: Output 0 1: Output 1 | 0 |
| 4 | MASKD4 | EPWM4 mask data 0: Output 0 1: Output 1 | 0 |
| 3 | MASKD3 | EPWM3 mask data 0: Output 0 1: Output 1 | 0 |
| 2 | MASKD2 | EPWM2 mask data 0: Output 0 1: Output 1 | 0 |
| 1 | MASKD1 | EPWM1 mask data 0: Output 0 1: Output 1 | 0 |
| 0 | MASKD0 | EPWM0 mask data 0: Output 0 1: Output 1 | 0 |

13.5.13 EPWM mask output control preset register (MASKNXT)

| Bit | Symbol | Description | Reset value |
|-------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:25 | - | Reserved | - |
| 24 | HALLEN | HALL detection mode enable bit 0: Disable 1: Enable | 0 |
| 23 | HALLCLR | HALL error status clear bit 0: Writing 0 is invalid 1: Writing 1 clears the HALLST error status and resets it to the initial state 000. Reading as 0. | 0 |
| | | Note 1: If an error state or sequence occurs when HALLST = 111, the HALL detection function stops. To enable HALL status again, write 1 to clear the 111 state. | |
| 22:20 | HALLST | HALL interface status bit (read-only) Detect the state corresponding to {CAP2, CAP1, CAP0}. 000: Status 0 (initial state) 001: Status 1 010: Status 2 011: Status 3 100: Status 4 101: Status 5 110: Status 6 111: Error status | 0x0 |
| | | Note 1: This status indicates the internal detection of the HALL interface in the chip, which can be used to determine if a valid state has been entered. If there are errors in the states of the three HALL sensors or errors in the order of the states, this status bit will be set to 111. Valid sequence 6-2-3-1-5-4-6- 1: Valid sequence 6-4-5-1-3-2-6- 2: Note 2: Under a valid status bit, if the enable mask preset data loading function is enabled, the corresponding mask preset cache data is loaded into the MASK register at the loading point. For example, when the HALL detection changes to state 3, the data of mask preset cache 3 is loaded into the MASK register at the first loading point after entering state 3. Note 3: Output the data of mask preset cache 7 under the initial state 000 or error state 111. | |
| 19 | - | Reserved | - |

| | | | |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 18:16 | PMASKSEL | <p>Mask preset cache selection bit;</p> <p>000: Select mask preset cache 0 001: Select mask preset cache 1 010: Select mask preset cache 2 011: Select mask preset cache 3 100: Select mask preset cache 4 101: Select mask preset cache 5 110: Select mask preset cache 6 111: Select mask preset cache 7</p> <p>Note 1: This selection bit affects the read and write of the lower 16 bits of data. There are 6 mask preset caches inside the EPWM.</p> <p>If it is 000: then the lower 16 bits of this register are read and written as the data in mask cache 0, If it is 001: then the lower 16 bits of this register are read and written as the data in mask cache 1, If it is 110: then the lower 16 bits of this register are read and written as the data in mask cache 6.</p> <p>Note 2: When HALLEN = 0, the data in mask preset cache 0 is loaded by default.</p> | 0x0 |
| 15:14 | - | Reserved | - |
| 13 | PMASKEN5 | EPWM5 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 12 | PMASKEN4 | EPWM4 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 11 | PMASKEN3 | EPWM3 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 10 | PMASKEN2 | EPWM2 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 9 | PMASKEN1 | EPWM1 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 8 | PMASKEN0 | EPWM0 mask output enable preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 7:6 | - | Reserved | - |
| 5 | PMASKD5 | EPWM5 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 4 | PMASKD4 | EPWM4 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 3 | PMASKD3 | EPWM3 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 2 | PMASKD2 | EPWM2 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |

| | | | |
|---|---------|------------------------------------------------------------------------------------------------------------------|---|
| 1 | PMASKD1 | EPWM1 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |
| 0 | PMASKD0 | EPWM0 mask data preset bit (This bit can be set to load into the MASK register at the loading point of EPWMn) | 0 |

13.5.14 EPWM trigger compare register (CMPTGD0-1)

| Bit | Symbol | Description | Reset value |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:20 | - | Reserved | - |
| 19 | CMPTGDSn | EPWM count comparator n trigger mode (Valid in center-aligned counting) 0: Triggered on count down 1: Triggered on count up | 0 |
| 18:16 | CMPPCHSn | EPWM digital comparator n compare channel selection 000: PWM0 counter 001: PWM1 counter 010: PWM2 counter 011: PWM3 counter 100: PWM4 counter 101: PWM5 counter Other value: PWM0 counter | 0x0 |
| 15:0 | CMPTGDn | EPWM count comparator n trigger compare value | 0x0 |

13.5.15 EPWM interrupt enable register (IMSC)

| Bit | Symbol | Description | Reset value |
|-------|--------------------|------------------------------------------------------------------------|-------------|
| 31 | EN_BRKIF | EPWM fault interrupt enable bit 0: Disable 1: Enable | 0 |
| 30 | EN_HALLIF | HALL status error interrupt enable bit 0: Disable 1: Enable | 0 |
| 29:24 | EN_DIFn (n=5-0) | EPWMn downward compare interrupt enable bit 0: Disable 1: Enable | 0x0 |
| 23:22 | - | Reserved | - |
| 21:16 | EN_UIFn (n=5-0) | EPWMn upward compare interrupt enable bit 0: Disable 1: Enable | 0x0 |
| 15 | EN_DC1IF | Count comparator 1 interrupt enable bit 0: Disable 1: Enable | 0 |
| 14 | EN_DC0IF | Count comparator 0 interrupt enable bit 0: Disable 1: Enable | 0 |
| 13:8 | EN_PIFn (n=5-0) | EPWMn period interrupt enable bit 0: Disable 1: Enable | 0x0 |
| 7:6 | - | Reserved | - |
| 5:0 | EN_ZIFn (n=5-0) | EPWMn zero interrupt enable bit 0: Disable 1: Enable | 0x0 |

13.5.16 EPWM interrupt source status register (RIS)

| Bit | Symbol | Description | Reset value |
|---------------|---------------------|--------------------------------------------------------------------------------------------------------------|-------------|
| 31 | RIS_BRKIF | EPWM fault interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0 |
| 30 | RIS_HALLIF | HALL status error interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0 |
| 29:24 | RIS_DIFn (n=5-0) | EPWMn downward compare interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0x0 |
| 23:22 | - | Reserved | - |
| 21:16 | RIS_UIFn (n=5-0) | EPWMn upward compare interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0x0 |
| 15 | RIS_DC1IF | Count comparator 1 interrupt status bit 0: Disable 1: Enable | 0 |
| 14 | RIS_DC0IF | Count comparator 0 interrupt status bit 0: Disable 1: Enable | 0 |
| 13:8 n=5-0 | RIS_PIFn (n=5-0) | EPWMn period interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0x0 |
| 7:6 | - | Reserved | - |
| 5:0 n=5-0 | RIS_ZIFn (n=5-0) | EPWMn zero point interrupt source status bit 0: No interrupt generated 1: Generated an interrupt | 0x0 |

13.5.17 EPWM enabled interrupt status register (MIS)

| Bit | Symbol | Description | Reset value |
|-------|---------------------|------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | MIS_BRKIF | EPWM fault enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0 |
| 30 | MIS_HALLIF | HALL state error enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0 |
| 29:24 | MIS_DIFn (n=5-0) | EPWMn downward compare enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0x0 |
| 23:22 | - | - | - |
| 21:16 | MIS_UIFn (n=5-0) | EPWMn upward compare enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0x0 |
| 15 | RIS_DC1IF | Counting comparator 1 enabled interrupt status bit 0: Disable 1: Enable | 0 |
| 14 | RIS_DC0IF | Counting comparator 0 enabled interrupt status bit 0: Disable 1: Enable | 0 |
| 13:8 | MIS_PIFn (n=5-0) | EPWMn period enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0x0 |
| 7:6 | - | - | - |
| 5:0 | MIS_ZIFn (n=5-0) | EPWMn zero point enabled interrupt status bit 0: No interrupt generated 1: An interrupt is enabled and generated | 0x0 |

13.5.18 EPWM interrupt clear control register (ICLR)

| Bit | Symbol | Description | Reset value |
|-------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | ICLR_BRKIF | EPWM fault interrupt clear control bit 0: No effect 1: Clear RIS_BRKIF flag bit | 0 |
| 30 | ICLR_HALLIF | HALL state error interrupt clear control bit 0: No effect 1: Clear RIS_HALLIF flag bit Note: RIS_HALLIF flag cannot be cleared if HALLST=111 | 0 |
| 29:24 | ICLR_DIFn (n=5-0) | EPWMn downward compare interrupt clear control bit 0: No effect 1: Clear RIS_DIFn flag bit | 0x0 |
| 23:22 | - | - | - |
| 21:16 | ICLR_UIFn (n=5-0) | EPWMn upward compare interrupt clear control bit 0: No effect 1: Clear RIS_UIFn flag bit | 0x0 |
| 15 | ICLR_DC1IF | Counting comparator 1 interrupt clear control bit 0: No effect 1: Clear RIS_DC1IF flag bit | 0 |
| 14 | ICLR_DC0IF | Counting comparator 0 interrupt clear control bit 0: No effect 1: Clear RIS_DC0IF flag bit | 0 |
| 13:8 | ICLR_PIFn (n=5-0) | EPWMn period interrupt clear control bit 0: No effect 1: Clear RIS_PIFn flag bit | 0x0 |
| 7:6 | - | - | - |
| 5:0 | ICLR_ZIFn (n=5-0) | EPWMn zero point interrupt clear control bit 0: No effect 1: Clear RIS_ZIFn flag bit | 0x0 |

13.5.19 EPWM interrupt accumulation control register (IFA)

| Bit | Symbol | Description | Reset value |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:12 | BRKIFCMP | Accumulated compare value for fault protection interrupt When the fault interrupt accumulates to (BRKIFCMP+1), set the BRKIF interrupt flag bit to 1. | 0x0 |
| 11:9 | - | Reserved | 0x0 |
| 8 | BRKIFAEN | Fault protection interrupt accumulation enable bit 0: Disable 1: Enable | 0 |
| 7:4 | ZIFCMP | Zero point interrupt accumulation compare value When the zero point interrupt of the corresponding channel accumulates to (ZIFCMP+1), set the ZIFn interrupt flag bit to 1(all channels set to the same compare value) | 0x0 |
| 3:1 | - | Reserved | - |
| 0 | ZIFAEN | Zero point interrupt accumulation enable bit 0: Disable 1: Enable | 0 |

13.5.20 EPWM write enable control register (LOCK)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0x55, enable the operation of other EPWM registers; When LOCK=0xaa, only enable the operation of the EPWM period register and the compare register. When LOCK=other values, disable the operation of EPWM related registers. | 0x0 |

13.5.21 EPWM fault protection recovery delay register (BRKRDT)

| Bit | Symbol | Description | Reset value |
|-------|--------|----------------------------------------------------------------------------------------------------------------|-------------|
| 31:20 | - | Must be set to 0 | 0x0 |
| 19:16 | FILS | Fault protection (brake) signal filter time selection | 0x0 |
| | | 0000: (0~1)* TPCLK | |
| | | 0001: (1~2)* TPCLK | |
| | | 0010: (2~3)* TPCLK | |
| | | 0011: (4~5)* TPCLK | |
| | | 0100: (8~9)* TPCLK | |
| | | 0101: (16~17)* TPCLK | |
| | | 0110: (24~25)* TPCLK | |
| | | 0111: (32~33)* TPCLK | |
| | | 1000: (48~49)* TPCLK | |
| | | 1001: (64~65)* TPCLK | |
| | | 1010: (80~81)* TPCLK | |
| | | 1011: (96~97)* TPCLK | |
| | | 1100: (112~113)* TPCLK | |
| | | Other: (0~1)* TPCLK | |
| 15:0 | RDT | Fault protection recovery delay (only available in delayed recovery mode) Delay time = $RDT \times TAPBCLK$ | 0x0 |



Chapter 14 Universal Asynchronous Receiver Transmitter (UART)

14.1 Overview

It contains 1 universal asynchronous serial interface.

14.2 Features

- ◆ Full duplex, asynchronous communication.
- ◆ Programmable serial interface features.
 - Data bit length can be set to 5-8 bits.
 - Parity bit can be set to odd, even, no parity, or fixed parity generation and detection.
 - Stop bit length can be set to 1, 1.5, or 2 bits.

14.3 Function description

14.3.1 UART function mode

UART is a full-duplex asynchronous communication interface. The UART transceiver contains a buffer for both transmitting and receiving, and the byte length and stop bit length can be flexibly set. Communication parameters for the full-duplex serial interface can be configured.

14.3.2 UART interrupts and status

UART supports three types of interrupts, including:

- Line status interrupts (parity check error, frame error, break interrupt).
- Receive data valid interrupt.
- Transmit holding register null interrupt.



14.4 Register mapping

(UART0 base address = 0x4006_4000)

RO: read only; WO: write only; R/W: read/write.

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------|-------------|
| RBR | 0x000 | RO | Receive Buffer Register | - |
| THR | 0x004 | WO | Transmit Buffer Register | - |
| DLR | 0x008 | R/W | Baud Rate Divider Register | 0x1 |
| IER | 0x00c | R/W | Interrupt Enable Register | 0x0 |
| IIR | 0x010 | RO | Interrupt Status Register | 0x1 |
| LCR | 0x018 | R/W | Line Control Register | 0x0 |
| MCR | 0x01C | R/W | Modem Control Register | 0x0 |
| LSR | 0x020 | RO | Line Status Register | 0x60 |
| END | 0x030 | W | UART Access End Register | - |



14.5 Register description

14.5.1 Receive buffer register (RBR)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | RBR | Read operation, returns the data received from the receive buffer. | - |

14.5.2 Transmit buffer register (THR)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | THR | Write data to the transmit buffer, the UART module will subsequently send the data out from the buffer. | - |

14.5.3 Baud rate divider register (DLR)

| Bit | Symbol | Description | Reset value |
|-------|--------|------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:0 | DLR | Baud rate = PCLK/ (16 x DLR) | 0x1 |

14.5.4 Interrupt enable register (IER)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------------------|-------------|
| 31:3 | - | Reserved | - |
| 2 | RLSIE | Receive line status interrupt enable bit 0: Disable 1: Enable | 0 |
| 1 | THREIE | Transmit holding register empty interrupt enable bit 0: Disable 1: Enable | 0 |
| 0 | RBRIE | Receive data valid interrupt/receive timer overflow interrupt enable bit 0: Disable 1: Enable | 0 |



14.5.5 Interrupt status register (IIR)

| Bit | Symbol | Description | Reset value |
|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:4 | - | Reserved | - |
| 3:1 | INTID | Interrupt status indication 0x0: Modem status has been changed. 0x1: Transmit holding register is empty. 0x2: Receive data is valid. 0x3: Receive line status | 0x0 |
| 0 | INT STATUS | Interrupt status 0: At least one interrupt in the queue. 1: No interrupts in the queue. | 1 |

14.5.6 Line control register (LCR)

| Bit | Symbol | Description | Reset value |
|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:7 | - | Reserved | - |
| 6 | BCON | Break control bit When this bit is written as 1, it enables Break transmission, and TXD port is forced to output logic 0. | 0 |
| 5:4 | PSEL | Parity bit selection 0x0: Odd parity, where odd numbers of logic 1 are transmitted and checked in each byte. 0x1: Even parity, where even numbers of logic 1 are transmitted and checked in each byte. 0x2: Parity bit is forced to 1. 0x3: Parity bit is forced to 0. | 0x0 |
| 3 | PEN | Parity check bit enable 0: Disable parity check bit generation and detection 1: Enable parity check bit generation and detection | 0 |
| 2 | SBS | Stop bit selection 0: 1-bit stop bit 1: When the transmit word length is 5 bits, the stop bit is 1.5 bits; when the transmit word length is other, the stop bit is 2 bits. | 0 |
| 1:0 | WLS | Word length select bit 0x0: 5-bit 0x1: 6-bit 0x2: 7-bit 0x3: 8-bit | 0x0 |



14.5.7 Modem control register (MCR)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------|-------------|
| 31:5 | - | Reserved | - |
| 4 | MLBM | Modem loopback mode 0: Disable Modem loopback mode 1: Enable Modem loopback mode | 0 |
| 3:0 | - | Reserved | - |

14.5.8 Line status register (LSR)

| Bit | Symbol | Description | Reset value |
|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:7 | - | Reserved | - |
| 6 | TEMT | Transmit buffer empty flag bit (read-only) 0: Transmit buffer has unsent data 1: Transmit buffer is empty | 1 |
| 5 | THRE | Transmit register empty flag bit (read-only) 0: Transmit register has unsent data 1: Transmit register is empty | 1 |
| 4 | BI | Break interrupt flag bit (read-only) 0: No break interrupt detected. 1: Break interrupt detected. A break interrupt is generated when the UART data input is held low during a transmission (start bit, data, parity bit, and stop bit). The UART remains idle until the data input goes high. The bit can be cleared by reading LSR. | 0 |
| 3 | FE | Frame error flag bit (read-only) 0: No frame error detected. 1: Frame error detected. The bit can be cleared by reading LSR. | 0 |
| 2 | PE | Parity check error flag bit (read-only) 0: No parity check error detected. 1: Parity check error detected. The bit can be cleared by reading LSR. | 0 |
| 1 | - | Reserved | 0 |
| 0 | RDR | Receiver data valid flag bit (read-only) 0: No unread data in receive buffer. 1: Unread data in receive buffer. | 0 |

14.5.9 UART access end register (END)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------------------|-------------|
| 31: 0 | END | UART access end register 0x0: Enable access to registers outside of UART. Other: Write disabled. | - |

Note:

1. After accessing UART-related registers, before operating registers outside of UART, the END register must be written as 0.
2. When operating the END register, if there is an interrupt that breaks the process, it may have an impact on the bus. To avoid errors, there are two ways to solve it:
 - (1) Disable interrupt enable before operating the END register, and enable interrupt after the operation is completed.
 - (2) Execute a write operation on the END register in the interrupt service program.

Chapter 15 I²C Serial Interface Controller (I²C)

15.1 Overview

I²C is a two-wire bi-directional serial bus that provides a simple and efficient connection for exchanging data between devices. I²C is a true multi-host bus, incorporating conflict detection and arbitration mechanisms. The conflict detection and arbitration mechanisms are used to prevent data corruption in the event that two or more hosts attempt to control the bus at the same time.

15.2 Features

- ◆ Support for master/slave mode.
- ◆ Bidirectional data transmission between master and slave devices.
- ◆ Multi-master bus.
- ◆ Arbitration of simultaneous data transmission between multiple masters to avoid serial data corruption on the bus.
- ◆ Use of serial synchronous clock for the bus, enabling devices to transmit at different rates.
- ◆ Serial synchronous clock can be used for handshaking to implement suspending and resuming serial transmission.
- ◆ Programmable clock for controlling multiple speeds.
- ◆ Support for 7-bit/10-bit slave address modes.
- ◆ Support for multiple address recognition (4 sets of slave addresses and 1 set of slave extended addresses with mask options).
- ◆ Support for wake-up mode.

15.3 Register mapping

(I2C0 base address = 0x4006_4300) RO: read only; WO: write only; R/W: read/write.

| Register | Offset value | R/W | Description | Reser value |
|----------|--------------|-----|---------------------------------------------------------|-------------|
| CONSET | 0x000 | R/W | I ² C Control Set Register | 0x0 |
| CONCLR | 0x004 | WO | I ² C Control Clear Register | 0x0 |
| STAT | 0x008 | RO | I ² C Status Register | 0xF8 |
| DAT | 0x00C | R/W | I ² C Data Register | 0x0 |
| CLK | 0x010 | R/W | I ² C Clock Control Register | 0x0 |
| ADR0 | 0x014 | R/W | I ² C Slave Address Register 0 | 0x0 |
| ADM0 | 0x018 | R/W | I ² C Slave Address Mask Register 0 | 0xFE |
| XADR0 | 0x01C | R/W | I ² C Extended Slave Address Register 0 | 0x0 |
| XADM0 | 0x020 | R/W | I ² C Extended Slave Address Mask Register 0 | 0x1FE |
| RST | 0x024 | WO | I ² C Software Reset Register | 0x0 |
| ADR1 | 0x028 | R/W | I ² C Slave Address Register 1 | 0x0 |
| ADM1 | 0x02C | R/W | I ² C Slave Address Mask Register 1 | 0xFE |
| ADR2 | 0x030 | R/W | I ² C Slave Address Register 2 | 0x0 |
| ADM2 | 0x034 | R/W | I ² C Slave Address Mask Register 2 | 0xFE |
| ADR3 | 0x038 | R/W | I ² C Slave Address Register 3 | 0x0 |
| ADM3 | 0x03C | R/W | I ² C Slave Address Mask Register 3 | 0xFE |

15.4 Register description

15.4.1 I²C control set register (CONSET)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:9 | - | Reserved | - |
| 8 | GCF | I ² C broadcast call flag bit Read-only 0: No broadcast call received 1: Broadcast call address matched This flag is cleared when a stop bit/restart bit/reset signal is received | 0 |
| 7 | I2CIE | Interrupt enable flag bit 0: Disable 1: Enable | 0 |
| 6 | I2CEN | I ² C interface enable flag bit 0: Disable I ² C interface 1: Enable I ² C interface Note: Enable I2C interface by writing 1 to the I2CEN bit, disable I2C interface by writing 1 to the I2CENC bit (I2CxCONCLR). Only valid in slave mode (address match unsuccessful). | 0 |
| 5 | STA | Start flag bit 1: I ² C enters master mode and sends a start signal; - If I ² C is already in master mode, it sends a restart signal - If I ² C is in slave mode, writing 1 will end the current transmission and wait for the bus to be idle before entering master mode. 0: No effect. - The flag is automatically cleared when the start or restart bit is sent. | 0 |
| 4 | STO | Stop flag bit Writing 1 in master mode sends a stop bit. Writing 1 in slave mode is treated as receiving a stop bit - When both STA and STO are set, the I ² C module sends a stop bit first, then sends a start bit - The flag is automatically cleared when the stop bit is sent. | 0 |
| 3 | SI | I ² C interrupt flag bit Read-only This flag is set when there is a bus status change in the I ² C It can be cleared by writing 1 to the SIC bit | 0 |
| 2 | AA | Acknowledge flag bit 0: No ACK signal received 1: Respond with an ACK signal in the following cases ● Slave address match ● Broadcasting enabled and broadcast address received ● Can be cleared by writing 1 to the AAC bit | 0 |

| | | | |
|---|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| | | when receiving data in master or slave mode | |
| 1 | XADRF | I ² C 10-bit Sslave address flag bit Read-only 0: I ² C address does not match 1: I ² C 10-bit address matched This flag is cleared when a stop bit/reset signal is received | 0 |
| 0 | ADRF | I ² C 7-bit slave address flag bit, read-only 0: I ² C address does not match 1: I ² C 7-bit address matched This flag is cleared when a stop bit/reset signal is received | 0 |

15.4.2 I²C control clear register (CONCLR)

| Bit | Symbol | Description | Reset value |
|------|--------|------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7 | I2CIEC | I ² C interrupt disable bit 0: Clear the I2CIE bit 1: No effect | 0 |
| 6 | I2CENC | I ² C interface disable bit 1: Clear the I2CEN bit 0: No effect | 0 |
| 5 | STAC | Start flag clear bit 0: No effect 1: Clear the STA bit | - |
| 4 | - | Reserved | |
| 3 | SIC | I ² C interrupt flag clear bit 0: No effect 1: Clear the SI bit | 0 |
| 2 | AAC | I ² C acknowledge flag clear bit 0: No effect 1: Clear the AA bit | 0 |
| 1:0 | - | Reserved | - |

Note: I²C operation requires the clearing of the corresponding flag bit to enter the next state.

15.4.3 I²C status register (STAT)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | Status | <p>I²C status code</p> <p>00H: Bus error (master mode only)</p> <p>08H: Start bit transmitted</p> <p>10H: Restart bit transmitted</p> <p>18H: Address + Write bit transmitted, ACK received</p> <p>20H: Address + Write bit transmitted, no ACK received</p> <p>28H: Data transmitted in master mode, ACK received</p> <p>30H: Data transmitted in master mode, no ACK received</p> <p>38H: Arbitration fail in address or data transmission</p> <p>40H: Address + Read bit transmitted, ACK received</p> <p>48H: Address + Read bit transmitted, no ACK received</p> <p>50H: Data received in master mode, ACK returned</p> <p>58H: Data received in master mode, no ACK returned</p> <p>60H: Address + Write bit received in slave mode, ACK returned</p> <p>68H: Arbitration fail in master mode, slave address + Write bit received, ACK returned</p> <p>70H: Received broadcast call address, ACK returned</p> <p>78H: Arbitration lost in master mode, broadcast call address received, ACK returned</p> <p>80H: Data received in slave mode after address match, ACK returned</p> <p>88H: Data received in slave mode after address match, no ACK returned</p> <p>90H: Data received in slave mode after broadcast call address match, ACK returned</p> <p>98H: Data received in slave mode after broadcast call address match, no ACK returned</p> <p>A0H: Stop or restart signal received in slave mode</p> <p>A8H: Address + Read bit received in slave mode, ACK returned</p> <p>B0H: Arbitration fail in master mode, slave address + Read bit received, ACK returned</p> <p>B8H: Data transmitted in slave mode, ACK received</p> <p>C0H: Data transmitted in slave mode, no ACK received</p> <p>C8H: Last data byte transmitted in slave mode, ACK received</p> <p>D0H: Last data byte transmitted in slave mode, no ACK received</p> <p>D8H: Not used</p> <p>E0H: Second address sent in master mode, ACK received</p> <p>E8H: Second address sent in master mode, no ACK received</p> <p>F0H: Not used</p> | 0xF8 |

| | | | |
|--|--|------------------------------------------|--|
| | | F8H: Uncertain status Other: Reserved | |
|--|--|------------------------------------------|--|

15.4.4 I²C data register (DAT)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | Data | Received data or data to be transmitted (reading received data is required immediately after data reception is completed). | 0x0 |

15.4.5 I²C clock control register (CLK)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------|-------------|
| 31:7 | - | Reserved | - |
| 6:4 | M | Sampling clock = PCLK/ ($2^M \times (N+1)$) | 0x0 |
| 3:0 | N | SCL clock = PCLK/ ($2^N \times (N+1) \times 10$) | 0x0 |

15.4.6 I²C slave address register (ADR0/ADR1/ADR2/ADR3)

| Bit | Symbol | Description | Reset value |
|------|---------|-----------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:1 | Address | Slave address | 0x0 |
| 0 | GC | 1: Enable broadcast call address recognition 0: Disable broadcast call address recognition | 0 |

15.4.7 I²C slave address mask register (ADM0/ADM1/ADM2/ADM3)

| Bit | Symbol | Description | Reset value |
|------|--------|---------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:1 | MASK | Mask bit 0: Do not compare the address of this bit 1: Compare the address of this bit | 0x7F |
| 0 | - | Reserved | - |

15.4.8 I²C extended slave address register (XADR0)

| Bit | Symbol | Description | Reset value |
|-------|---------|-----------------------------------------------------------------------------------------------|-------------|
| 31:11 | - | Reserved | - |
| 10:1 | Address | 10-bit slave address | 0x0 |
| 0 | GC | 1: Enable broadcast call address recognition 0: Disable broadcast call address recognition | 0 |

15.4.9 I²C extended slave address mask register (XADM0)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------|-------------|
| 31:9 | - | Reserved | - |
| 8:1 | MASK | Mask bit 0: The address of this bit is not compared 1: The address of this bit is compared | 0xFF |
| 0 | - | Reserved | - |

15.4.10 I²C software reset register (RST)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | RST | Write any value to generate a software reset. | 0x0 |

Chapter 16 SPI Controller (SSP/SPI)

16.1 Overview

Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol that operates in full-duplex mode. Devices can operate in master/slave mode and communicate with each other using a 4-wire bidirectional interface. SPI performs serial-to-parallel conversion when receiving data from a peripheral device and parallel-to-serial conversion when sending data to a peripheral device. The SPI controller can be configured as either a master or a slave device.

16.2 Features

- ◆ Support master or slave mode.
- ◆ Full-duplex.
- ◆ Configurable bit length for transmission (4-bit to 16-bit).
- ◆ MSB first for transmission/reception.
- ◆ Built-in a receive buffer and a transmit buffer.

16.3 Register mapping

(SSP0 base address= 0x4006_3000) RO: Read only, WO: Write Only, R/W: Read/Write.

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------------------|-------------|
| CON | 0x000 | R/W | SSP Control Register | 0x0 |
| STAT | 0x004 | RO | SSP Status Register | 0x3 |
| DAT | 0x008 | R/W | SSP Data Register | 0x0 |
| CLK | 0x00C | R/W | SSP Clock Control Register | 0x0 |
| IMSC | 0x010 | R/W | SSP Interrupt Enable Register | 0x0 |
| RIS | 0x014 | RO | SSP Interrupt Source Status Register | 0x8 |
| MIS | 0x018 | RO | SSP Enabled Interrupt Status Register | 0x0 |
| ICLR | 0x01C | WO | SSP Interrupt Clear Register | 0x0 |
| CSCR | 0x028 | R/W | SSP Software Chip Select Signal Register | 0x0 |

16.4 Register description

16.4.1 SSP control register (CON)

| Bit | Symbol | Description | Reset value |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:12 | - | Reserved | - |
| 11 | LBM | Loopback mode enable bit 0: Normal operation mode 1: Loopback mode, connect serial input to serial output | 0 |
| 10 | SSPEN | SSP enable bit 0: Disable 1: Enable | 0 |
| 9 | MS | Master/slave mode select bit 0: Master mode 1: Slave mode | 0 |
| 8 | | Reserved | 0 |
| 7 | CPH | Clock phase control bit 0: SSP samples data on the first clock edge 1: SSP samples data on the second clock edge | 0 |
| 6 | CPO | Clock output polarity select bit 0: SPI_CLK is low when idle 1: SPI_CLK is high when idle | 0 |
| 5:4 | FRF | Frame format 0x0: SPI-compatible frame format 0x1: TI/SS-compatible frame format 0x2: Microwire-compatible frame format 0x3: Reserved | 0x0 |
| 3:0 | DSS | Data transfer length select bit 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: 4-bit 0x4: 5-bit 0x5: 6-bit 0x6: 7-bit 0x7: 8-bit 0x8: 9-bit 0x9: 10-bit 0xA: 11-bit 0xB: 12-bit 0xC: 13-bit 0xD: 14-bit 0xE: 15-bit 0xF: 16-bit | 0x0 |

16.4.2 SSP status register (STAT)

| Bit | Symbol | Description | Reset value |
|------|--------|------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:5 | - | Reserved | - |
| 4 | BSY | Busy flag bit, read-only 0: SSP is idle 1: SSP is transmitting/receiving data or Transmit Buffer has been written data | 0 |
| 3:0 | | Reserved | 0x3 |

16.4.3 SSP data register (DAT)

| Bit | Symbol | Description | Reset value |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:0 | DATA | When writing data to this register, the data will be written into the transmit register and sent out when there is no data being transmitted on the bus. If there is data being transmitted on the bus, the data will be stored in the buffer and sent after the previous transmission is completed. The minimum interval between two transmissions is 3 SSPCLK clocks. When the data length is less than 16 bits, it needs to be right-aligned. When reading this register, the most recently received data is read, and when the length of the data is less than 16 bits, it should be right-aligned. | 0x0 |

16.4.4 SSP clock controller (CLK)

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:8 | M | $SSPCLK = PCLK / ((M+1) \times N)$ N is an even number from 2 to 254 | 0x0 |
| 7:0 | N | | 0x0 |

16.4.5 SSP interrupt enable register (IMSC)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:4 | - | Reserved | - |
| 3 | TXIM | Transmit Buffer interrupt enable bit 0: Disable interrupt for empty transmit buffer. 1: Enable interrupt for empty transmit buffer. | 0 |
| 2 | RXIM | Receive Buffer interrupt enable bit 0: Disable interrupt for received data in receive buffer. 1: Enable interrupt for received data in receive buffer. | 0 |
| 1 | RTIM | Receive Buffer timer overflow interrupt enable bit 0: Disable interrupt for receive buffer timer overflow. 1: Enable interrupt for receive buffer timer overflow. (Overflow time is 32 × SSPCLK) | 0 |
| 0 | RORIM | Receive Buffer overflow interrupt enable bit 0: Disable interrupt for receive buffer overflow. 1: Enable interrupt for receive buffer overflow. | 0 |

16.4.6 SSP interrupt source status register (RIS)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:4 | - | Reserved | - |
| 3 | TXRIS | Set when the transmit buffer is empty or data in the transmit buffer has been sent. (Automatically cleared when there is data in the transmit buffer) | 1 |
| 2 | RXRIS | Set when the receive buffer receives data. (Automatically cleared when there is no data in the receive buffer or data in the receive buffer has been read) | 0 |
| 1 | RTRIS | Set when the receive buffer receives data and remains unread for a timeout period. (Cleared by reading the data register or writing to the ICLR register) | 0 |
| 0 | RORRIS | When the receive buffer receives data and remains unread, and another frame of data is received, this bit is set, and the new data will be lost. (Cleared by writing to the ICLR register) | 0 |

16.4.7 SSP enabled interrupt status register (MIS)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------|-------------|
| 31:4 | - | Reserved | - |

| | | | |
|---|--------|------------------|---|
| 3 | TXMIS | = TXIM & TXRIS | 0 |
| 2 | RXMIS | = RXIM & RXRIS | 0 |
| 1 | RTMIS | = RTIM & RTRIS | 0 |
| 0 | RORMIS | = RORIM & RORRIS | 0 |

16.4.8 SSP interrupt clear register (ICLR)

| Bit | Symbol | Description | Reset value |
|------|--------|------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | RTIC | 1: Clear the RTRIS flag bit | 0 |
| 0 | RORIC | 1: Clear the RORRIS flag bit | 0 |

16.4.9 SSP software chip select signal register (CSCR)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:4 | - | Reserved | - |
| 3 | SWCS | Software chip select signal control bit in master mode 0: Output low level 1: Output high level | 0 |
| 2 | SWSEL | Chip select signal selection in master mode 0: The chip select signal is automatically controlled by the SPI module 1: The chip select signal is controlled by the SWCS bit | 0 |
| 1:0 | - | Reserved | - |

Chapter 17 Analog-to-Digital Conversion (ADC)

17.1 Overview

The chip contains a 12-bit, 23-channel fast successive approximation analog-to-digital converter (ADC).

17.2 Features

- ◆ Simulation input voltage range: $V_{SS} \sim V_{DD}$.
- ◆ Maximum sampling rate: 1.2Msps.
- ◆ Up to 23 single-ended analog input channels.
- ◆ Supports two power modes: high-speed mode and low-current mode.
- ◆ In high-speed mode, the conversion time for a single sample is $52 \cdot T_{ADCK}$ (sampling time set to $13.5 \cdot T_{ADCK}$).
- ◆ Single mode: performs one A/D conversion on a specified channel.
- ◆ Continuous mode: performs A/D conversions on all selected channels.
- ◆ Supports external input signal triggering of ADC conversion.
- ◆ Generates an interrupt when conversion is completed.
- ◆ Built-in AD conversion result comparator.
- ◆ The conversion results for each channel are stored in their corresponding data registers.

17.3 Functional description

17.3.1 ADC channels

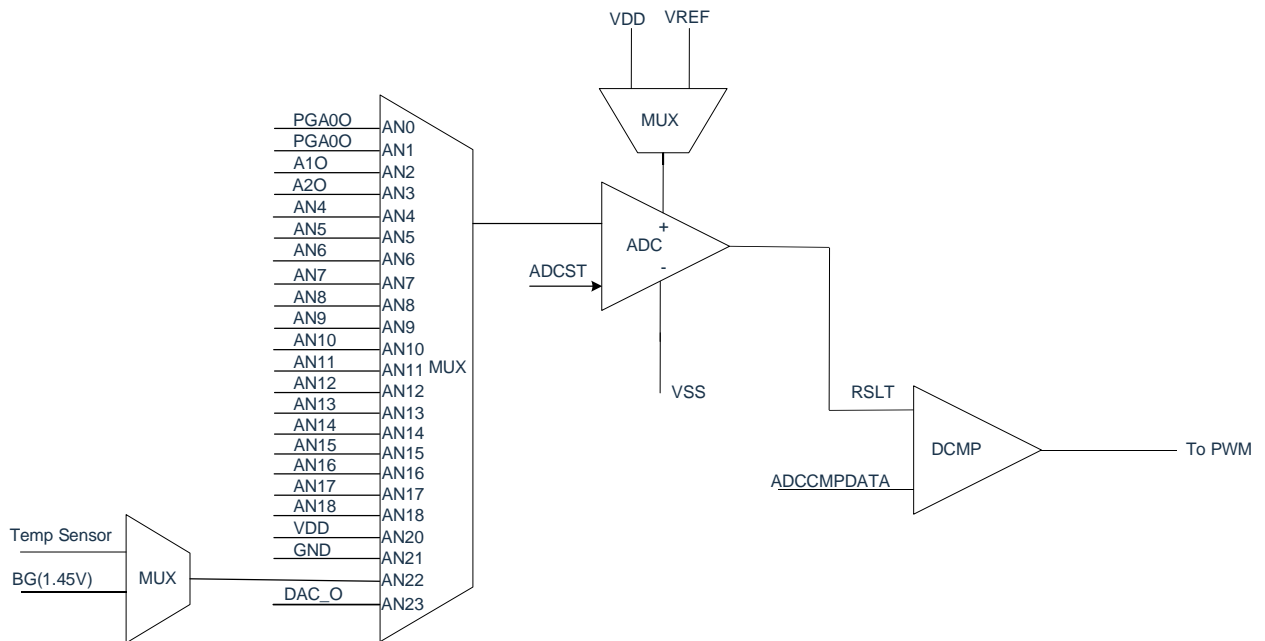
| ADC channel number (supports hardware trigger) | ADC channel | ADC channel priority | Description |
|------------------------------------------------|--------------|----------------------|-------------------------------------------|
| 0 | AN0 (PGA0O) | Highest | PGA0 channel (see Chapter 18 for details) |
| 1 | AN1 (PGA0O) | - | PGA0 channel (see Chapter 18 for details) |
| 2 | AN2 (A1O) | - | PGA1 channel (see Chapter 18 for details) |
| 3 | AN3 (A2O) | - | PGA2 channel (see Chapter 18 for details) |
| 4 | AN4 (P20) | - | External channel 4 |
| 5 | AN5 (P21) | - | External channel 5 |
| 6 | AN6 (P22) | - | External channel 6 |
| 7 | AN7 (P23) | - | External channel 7 |
| 8 | AN8 (P00) | - | External channel 8 |
| 9 | AN9 (P01) | - | External channel 9 |
| 10 | AN10 (P02) | - | External channel 10 |
| 11 | AN11 (P03) | - | External channel 11 |
| 12 | AN12 (P04) | - | External channel 12 |
| 13 | AN13 (P05) | - | External channel 13 |
| 14 | AN14 (P24) | - | External channel 14 |
| 15 | AN15 (P25) | - | External channel 15 |
| 16 | AN16 (P26) | - | External channel 16 |
| 17 | AN17 (P27) | - | External channel 17 |
| 18 | AN18 (P16) | - | External channel 18 |
| 19 | - | - | Disable selection |
| 20 | AN20 (VDD) | - | |
| 21 | AN21 (GND) | - | |
| 22 | AN22 (BG2AD) | - | BG1.45V/Temperature sensor channel |
| 23 | AN23 | Lowest | Internal channel |

Note: Any combination of AN0-AN23 channels supports continuous mode switching.

Internal channels of the ADC

| ADC internal channel number | ADC internal channel | Description |
|-----------------------------|----------------------|-------------------------------------|
| 1-3 | - | Disable selection |
| 4 | IAN_4 (DAC_O) | DAC output channel (see Chapter 20) |

17.3.2 Block diagram of ADC structure



17.3.3 ADC power consumption modes

There are two modes of ADC operation: high-speed mode and low current mode.

High-speed mode: This mode has a faster conversion speed.

Low current mode: This mode has a slightly slower conversion speed, and the operating current of the ADC is significantly reduced. This mode can be used to reduce the power consumption of the ADC for applications that do not require high conversion rates. The successive comparison time in this mode is 10 TADCKs longer than the high-speed mode.

17.3.4 ADC conversion modes

ADC conversion modes can be divided into two types: single conversion mode and continuous conversion mode.

Single conversion mode:

Performs a single conversion on the highest priority enabled channel, then finishes the operation and sets an interrupt flag.

Continuous conversion mode:

Performs conversions on all enabled channels, then finishes the operation and sets an interrupt flag. Disabled channels are ignored and skipped.

When ADCSWCHE=0, software channel switching is disabled, and the selection and enabling of ADC channels are controlled automatically by hardware.

When ADCSWCHE=1, software channel switching is enabled, and the selection and enabling of ADC channels are controlled by ADCSWCHS. After selecting a channel with ADCSWCHS, that channel is automatically enabled (ADCEN must be 1). In this case, both single and continuous modes perform conversions on the selected channel.

17.3.5 ADC clock

The ADC clock is derived from the APB clock and can be divided into eight different frequencies: 1/2/4/8/16/32/64/128, configured by ADCCON.ADCDIV.

In high-speed mode, the time for a single conversion (TADC) in single conversion mode is:

$2 \times \text{TADCK}$ (default settling time) + $13.5 \times \text{TADCK}$ (default sampling time) + $31.5 \times \text{TADCK}$ (successive comparison time) + $5 \times \text{TADCK}$

In high-speed mode, the time for completing one ADC conversion in continuous conversion mode (TADC) is:

$2 \times \text{TADCK}$ (default settling time) + $13.5 \times \text{TADCK}$ (default sampling time) + $31.5 \times \text{TADCK}$ (successive comparison time) + $3 \times \text{TADCK}$

When ADCSWCHE=1, the actual settling time is the time from selecting a channel to starting the conversion.

17.3.6 ADC channel selection and interrupt generation

| ADCCON (ADCSWCHE) | ADCSCAN (ADEn) | ADCCON (ADCMS) | Channel description | Result storage | Interrupt generation |
|----------------------|-------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | 0 | 0/1 | Disable all channels | - | - |
| 0 | 1 | 0 | Converts the highest-priority channel already enabled in the SCAN | After the conversion of a single channel is completed, the result is stored in the result register corresponding to the converted channel. | After the conversion of a single channel is completed, an interrupt is generated in the interrupt source corresponding to the converted channel (ADCRISn). |
| | | 1 | Converts all enabled channels in the SCAN in descending order of priority. | | |
| 1 | X | 0 | Converts the channel set in ADCSWCHS once | The result is in the result register corresponding to the channel with the highest SCAN enabled priority. | After the conversion is completed, the interrupt is generated in the interrupt source corresponding to the channel with the highest priority enabled by SCAN (ADCRISn). |
| | | 1 | SCAN enables as many active channels as it takes to convert, and the converted channel is always the channel set by ADCSWCHS. | After the conversion of a single channel is completed, the results are stored in the result registers corresponding to the SCAN-enabled channels in descending order of priority. | After the conversion of a single channel is completed, the interrupts are generated in the interrupt source corresponding to the SCAN-enabled channel in descending order of priority (ADCRISn). |

Note: If X is 0, the conversion of the channels set in ADCSWCHS will still be started, but the results and interrupts will not be updated in any registers.

17.3.7 ADC software start

Write 1 to the ADCCON2.ADCST bit to start the ADC conversion. The bit is automatically cleared by the hardware after the conversion is completed.

During the ADC conversion, any software or hardware trigger start signals will be ignored.

17.3.8 ADC hardware trigger start

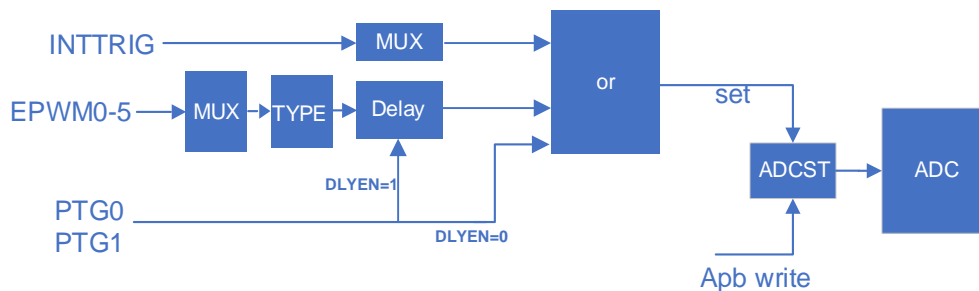
Trigger sources:

In addition to software-triggered conversion, the ADC can also be triggered by hardware signals. There are several types of hardware trigger sources:

- 1) Internal triggers
- 2) EPWM output channel triggers
- 3) EPWM count comparator 0 triggers
- 4) EPWM count comparator 1 triggers

Different types of trigger sources can be active simultaneously, and each type may have different trigger signals. For example, in EPWM output channel triggers, you can select one of EPWM0-EPWM5 as the trigger signal.

Figure 17-1: ADC hardware trigger start



Internal triggers:

Internal triggers include ADC, ACMP0, ACMP1, TIMER0/1 triggers.

ADC: ADC conversion completion

ACMP0: Event output of ACMP0

ACMP1: Event output of ACMP1

Timer0: Enabled interrupt of Timer0 (TMR0MIS)

Timer1: Enabled interrupt of Timer1 (TMR1MIS)

EPWM output channel triggers:

EPWM output channel triggers can be triggered by rising edge, falling edge, zero point, or period point to start ADC conversion. If an EPWM trigger signal is detected, you can choose to start the ADC conversion after a certain delay. If the output channels of EPWM are remapped, the EPWM trigger signal refers to the signal before remapping, i.e., IPGn signal.

EPWM output channel triggers only support hardware-selected channels and can set separate ADC conversion channels. That means, after the EPWM output channel trigger signal is generated, the conversion will be performed according to the configured channel. The conversion channel for ADC triggered by EPWM output channels is set in the ADCCHEPWM register. After the conversion is completed, it will revert to the channel settings in ADCSCAN.

EPWM Count Comparator Trigger:

EPWM count comparator 0/1 triggers can be set to start ADC conversion at any time within the EPWM period, similar to EPWM channel triggers. It is also possible to choose to start ADC conversion after a certain delay.

EPWM count comparator 0/1 triggers only support hardware-selected channels and can set separate ADC conversion channels. This means that after the trigger signal is generated, the conversion will be performed according to the configured channel. The conversion channel for ADC triggered by EPWM count comparator 0 is set in the ADCCHEPTG0 register, while the conversion channel for ADC triggered by EPWM count comparator 1 is set in the ADCCHEPTG1 register. After the conversion is completed, it will restore the channel settings in ADCSCAN.

EPWM trigger delay:

The ADCEPWMTGDLY register determines the delay time for EPWM trigger to start ADC conversion: $(ADCEPWMTGDLY[9:0]+3)*PCLK$

The range of EPWM trigger delay is as follows:

| pclk 48MHz (delay range) | pclk 64MHz (delay range) |
|--------------------------|--------------------------|
| 0.041us~21.34us | 0.031us~16.03us |
| 0.02us~31.32us | 0.015US~16.01us |

If $ADCEPWMTGDLY = 0$, EPWM comparator 0/EPWM comparator 1/rising edge/falling edge/period point/zero point will trigger ADC conversion with a delay of 3 PCLK clocks.

EPWM trigger start ADC configuration

EPWM-triggered ADC conversion has special timing requirements in some applications. In response to this demand, the ADC internally supports different EPWM trigger conditions that can be set with independent conversion channels. For example:

EPWM output channel trigger can select AN0, AN1, AN2 channels for conversion.

EPWM comparator 0 trigger can select AN17 channel for conversion.

EPWM comparator 1 trigger can select AN18 channel for conversion.

The channels selected for software start or other trigger start are AN5, AN6, AN7, AN8.

When there is no EPWM triggers, the default conversion channels are AN5-AN8.

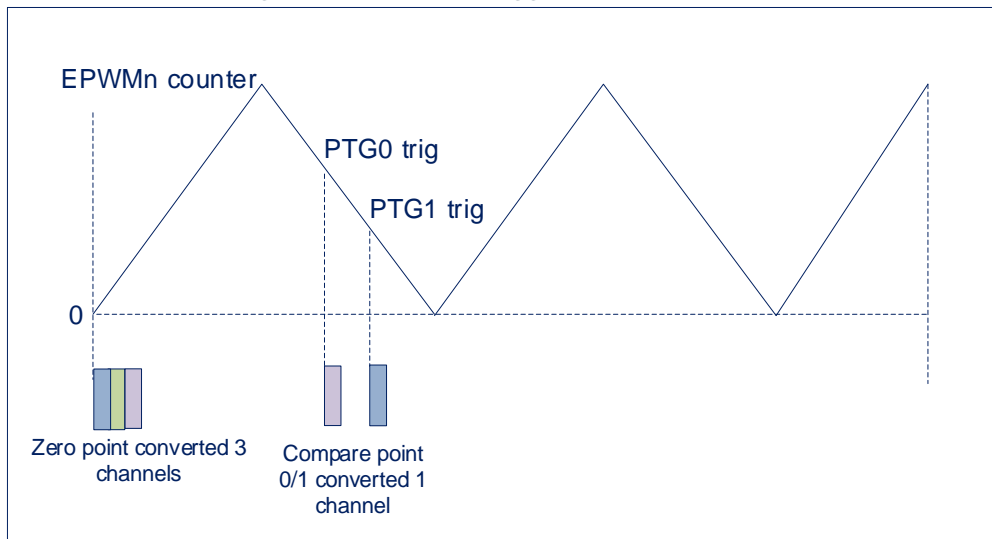
If EPWM's output channel trigger is activated, only channels AN0-AN2 will be selected for ADC conversion, and after conversion, it will automatically switch to channels AN5-AN8.

If EPWM's comparator 0 trigger is activated, only channel AN17 will be selected for ADC conversion, and after conversion, it will automatically switch to channels AN5-AN8.

If EPWM's comparator 1 trigger is activated, only channel AN18 will be selected for ADC conversion, and after conversion, it will automatically switch to channels AN5-AN8.

It should be noted that during the ADC conversion process, any other trigger signals will be ignored.

Figure 17-2: EPWM trigger to initiate ADC setup



Note 1: The channel that enables ADC conversion triggered at the zero point is determined by ADCCHPEM.

Note 2: The channel for ADC conversion enabled by Comparator 0 trigger is determined by ADCCHPTG0.

Note 3: The channel for ADC conversion enabled by Comparator 1 trigger is determined by ADCCHPTG1.

The channels for ADC conversion enabled by other triggering methods are determined by ADCSCAN or ADCSWCHS.

17.4 Register mapping

(ADC base address = 0x4006_8000) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|---------------------------|--------------|-----|-----------------------------------------------------------|-------------|
| CON _(P1B) | 0x000 | R/W | ADC Control Register | 0xD0000 |
| CON2 _(P1B) | 0x004 | R/W | ADC Control Register 2 | 0x0 |
| HWTG _(P1B) | 0x008 | R/W | ADC Hardware Trigger Control Register | 0x0 |
| PWMTGDLY _(P1B) | 0x00C | R/W | ADC EPWM Trigger Delay Data Register | 0x0 |
| SCAN _(P1B) | 0x010 | R/W | ADC Scan Register | 0x0 |
| CMP0 _(P1B) | 0x014 | R/W | ADC Comparator 0 Control Register | 0x0 |
| CMP1 _(P1B) | 0x018 | R/W | ADC Comparator 1 Control Register | 0x0 |
| IMSC _(P1B) | 0x01C | R/W | ADC Interrupt Enable Register | 0x0 |
| RIS | 0x020 | RO | ADC Interrupt Source Status Register | 0x0 |
| MIS | 0x024 | RO | ADC Enabled Interrupt Status Register | 0x0 |
| ICLR | 0x028 | WO | ADC Interrupt Clear Register | 0x0 |
| LOCK | 0x02C | R/W | ADC Write Enable Control Register | 0x0 |
| CHEPWM _(P1B) | 0x030 | R/W | ADC EPWM Output Trigger Conversion Channel Register | 0x0 |
| CHPTG0 _(P1B) | 0x034 | R/W | ADC EPWM Comparator 0 Trigger Conversion Channel Register | 0x0 |
| CHPTG1 _(P1B) | 0x038 | R/W | ADC EPWM Comparator 1 Trigger Conversion Channel Register | 0x0 |
| -- | -- | -- | -- | -- |
| DATA0 | 0x080 | RO | ADC Channel 0 Conversion Result Register | 0x0 |
| DATA1 | 0x084 | RO | ADC Channel 1 Conversion Result Register | 0x0 |
| DATA2 | 0x088 | RO | ADC Channel 2 Conversion Result Register | 0x0 |
| DATA3 | 0x08C | RO | ADC Channel 3 Conversion Result Register | 0x0 |
| DATA4 | 0x090 | RO | ADC Channel 4 Conversion Result Register | 0x0 |
| DATA5 | 0x094 | RO | ADC Channel 5 Conversion Result Register | 0x0 |
| DATA6 | 0x098 | RO | ADC Channel 6 Conversion Result Register | 0x0 |
| DATA7 | 0x09C | RO | ADC Channel 7 Conversion Result Register | 0x0 |
| DATA8 | 0x0A0 | RO | ADC Channel 8 Conversion Result Register | 0x0 |
| DATA9 | 0x0A4 | RO | ADC Channel 9 Conversion Result Register | 0x0 |
| DATA10 | 0x0A8 | RO | ADC Channel 10 Conversion Result Register | 0x0 |
| DATA11 | 0x0AC | RO | ADC Channel 11 Conversion Result Register | 0x0 |
| DATA12 | 0x0B0 | RO | ADC Channel 12 Conversion Result Register | 0x0 |
| DATA13 | 0x0B4 | RO | ADC Channel 13 Conversion Result Register | 0x0 |
| DATA14 | 0x0B8 | RO | ADC Channel 14 Conversion Result Register | 0x0 |

| | | | | |
|--------|-------|----|-------------------------------------------|-----|
| DATA15 | 0x0BC | RO | ADC Channel 15 Conversion Result Register | 0x0 |
| DATA16 | 0x0C0 | RO | ADC Channel 16 Conversion Result Register | 0x0 |
| DATA17 | 0x0C4 | RO | ADC Channel 17 Conversion Result Register | 0x0 |
| DATA18 | 0x0C8 | RO | ADC Channel 18 Conversion Result Register | 0x0 |
| -- | -- | -- | -- | - |
| DATA20 | 0x0D0 | RO | ADC Channel 20 Conversion Result Register | 0x0 |
| DATA21 | 0x0D4 | RO | ADC Channel 21 Conversion Result Register | 0x0 |
| DATA22 | 0x0D8 | RO | ADC Channel 22 Conversion Result Register | 0x0 |
| DATA22 | 0x0DC | RO | ADC Channel 23 Conversion Result Register | 0x0 |

Note:

1 The registers marked with (P1B) are protected registers.

When (P1B)LOCK=55H, the marked registers are allowed to be written; for other values, writing is prohibited

17.5 Register description

17.5.1 ADC control register (CON)

| Bit | Symbol | Description | Reset value |
|-------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | ADCRST | ADC module reset control bit 0: --- 1: ADC module reset | 0 |
| 30:26 | - | Reserved, must be set to 0 | 0x0 |
| 25:24 | ADMODE10 | ADC power mode select bit 00: High-speed mode 01: Reserved, disable selection. 10: Reserved, disable selection. 11: Low current mode | 0x0 |
| 23:16 | ADCNSMP | ADC internal sample time select bit 00000000: Disable selection 00000100: 5.5 ADC clock cycles 00000110: 6.5 ADC clock cycles 00000111: 7.5 ADC clock cycles 00001000: 8.5 ADC clock cycles 00001001: 9.5 ADC clock cycles 00001010: 10.5 ADC clock cycles 00001011: 11.5 ADC clock cycles 00001100: 12.5 ADC clock cycles 00001101: 13.5 ADC clock cycles 00001000: - 11111110: 254.5 ADC clock cycles 11111111: 255.5 ADC clock cycles | 0xD |
| 15 | - | Reserved | 0 |
| 14 | - | Reserved | 0 |
| 13 | ADCSWCHE | ADC channel software enable bit 0: Automatically turned on by hardware 1: The channel activation is determined by ADCSWCHS. | 0 |
| 12 | ADCNDISEN | ADC charge/discharge function select bit 0: Discharge 1: Charge | 0 |
| 11:8 | ADCNDISTS | ADC charge/discharge time select bit 0000: No charging or discharging 0001: Disable selection 0010: 2 ADC clockcycles 0011: 3 ADC clockcycles ... 1111: 15 ADC clockcycles | 0x0 |
| 7:6 | ADCVS | ADC positive reference select bit 00: Select VDD | 0x0 |

| | | | |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| | | 01: Select VREF 10: Reserved 11: Disable selection | |
| 5 | - | Reserved, must be set to 0 | 0 |
| 4 | ADCEN | ADC enable control bit 0: Disable 1: Enable | 0 |
| 3 | ADCMS | ADC conversion mode select bit 0: Single conversion 1: Continuous conversion (Convert all enabled ADC channels at one time, the order is channel 0 to channel 23, the hardware automatically ignores the channels that are not enabled, and no conversion operation will be generated.) | 0 |
| 2:0 | ADCDIV | ADC clock prescaler select bit $F_{ADC} = PCLK/2^{ADCDIV}$ | 0x0 |

17.5.2 ADC control register 2 (CON2)

| Bit | Symbol | Description | Reset value |
|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15:13 | ADCICHES | ADC internal channel (AN23) select bit 100: Select internal channel 4 Other: Disable selection | 0x0 |
| 12 | ADCSF4 | ADC conversion status flag bit 4 (read-only) 0: - 1: Single conversion completed | 0 |
| 11 | ADCSF3 | ADC conversion status flag bit 3 (read-only) 0: - 1: Two ADC clock cycles before conversion is completed | 0 |
| 10 | ADCSF2 | ADC conversion status flag bit 2 (read-only) 0: - 1: Two ADC clock cycles before conversion is completed | 0 |
| 9 | ADCSF1 | ADC conversion status flag bit 1 (read-only) 0: - 1: During conversion | 0 |
| 8 | ADCSF0 | ADC conversion status flag bit 1 (read-only) 0: - 1: During sampling | 0 |
| 7 | ADCST | ADC conversion starts (hardware automatically clears after conversion) Conversion finished or ADC in idle mode (Write 0 is invalid) 0: - 1: Start conversion (ADCEN must be 1) | 0 |
| 6 | ADCSMPWAIT | ADC sample time extension control bit 0: - 1: Forced hold sampling state during sampling | 0 |
| 5 | BG2ADSEL | ADC TS channel (AN22) select bit 0: TS temperature sensor 1: BG reference voltage 1.45V | 0 |
| 4:0 | ADCSWCHS | ADC channel software selection bit (Valid only when ADCSWCHE=1) Note: 10011 disable selection 00000: Select channel 0 00001: Select channel 1 10111: Select channel 23 Other: Reserved | 0x0 |

17.5.3 ADC hardware trigger control register (HWTG)

| Bit | Symbol | Description | Reset value |
|-------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15 | ADCINTTGEN | ADC internal function trigger enable bit 0: Disable 1: Enable | 0 |
| 14:12 | ADCINTTGSS | ADC internal function trigger source channel select bit 000: Reserved 001: ADC conversion end signal 010: ACMP0 event 011: ACMP1 event 100: Timer0 interrupt signal 101: Timer1 interrupt signal | 0x0 |
| 11 | ADCPTG1DLYEN | ADC EPWM count comparator 1 delay trigger enable bit 0: Enable 1: Disable (without delay) | - |
| 10 | ADCPTG0DLYEN | ADC EPWM count comparator 0 delay trigger enable bit 0: Enable 1: Disable (without delay) | - |
| 9 | ADCPTG1EN | ADC EPWM count comparator 1 trigger enable bit 0: Disable 1: Enable | 0 |
| 8 | ADCPTG0EN | ADC EPWM count comparator 0 trigger enable bit 0: Disable 1: Enable | 0 |
| 7 | ADCEPWMTEN | ADC EPWM output trigger enable bit 0: Disable 1: Enable | 0 |
| 6:4 | ADCEPWMTSS | ADC EPWM output trigger source channel select bit 000: The trigger source is EPWM0 001: The trigger source is EPWM1 010: The trigger source is EPWM2 011: The trigger source is EPWM3 100: The trigger source is EPWM4 101: The trigger source is EPWM5 11x: Reserved | 0x0 |
| 3:2 | - | Reserved | - |
| 1:0 | ADCPEWMTPS | ADC EPWMn trigger mode select bit (n=0-5) 00: Rising edge of the EPWMn waveform 01: EPWMn period point (IPGn) 10: Falling edge of the EPWMn waveform 11: Zero point of EPWMn (IPGn) | 0x0 |

17.5.4 ADC EPWM trigger delay register (EPWMTGDLY)

| Bit | Symbol | Description | Reset value |
|-------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:10 | - | Reserved | - |
| 9:0 | ADCEPWMTGDLY | ADC EPWM trigger delay data EPWM (including output channel triggering and EPWM comparator 0/1 triggering) delay trigger ADC delay data (see section 17.3.8 EPWM trigger delay for details) | 0x0 |

Note: The EPWMTGDLY register bit12 needs to be written 1 after the chip is powered on.

17.5.5 ADC scan register (SCAN)

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------|-------------|
| 31:24 | - | Reserved | - |
| 23:0 | ADCEn | ADC channel n enable bit (n=23-0,n≠19) 0: Disable 1: Enable | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.6 ADC EPWM output trigger conversion channel enable register (CHEPWM)

| Bit | Symbol | Description | Reset value |
|-------|------------|---------------------------------------------------------------------------------------------------|-------------|
| 31:24 | - | Reserved | - |
| 23:0 | ADCCHEPWMn | ADC EPWM output trigger conversion channel enable bit (n=23-0,n≠19) 0: Disable 1: Enable | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.7 ADC EPWM comparator 0 trigger conversion channel enable register (CHPTG0)

| Bit | Symbol | Description | Reset value |
|-------|------------|------------------------------------------------------------------------------------------------------|-------------|
| 31:24 | - | Reserved | - |
| 23:0 | ADCCHPTG0n | ADC EPWM comparator 0 trigger conversion channel enable bit (n=23-0,n≠19) 0: Disable 1: Enable | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.8 ADC EPWM comparator 1 trigger conversion channel enable register (CHPTG1)

| Bit | Symbol | Description | Reset value |
|-------|------------|------------------------------------------------------------------------------------------------------|-------------|
| 31:24 | - | Reserved | - |
| 23:0 | ADCCHPTG1n | ADC EPWM comparator 1 trigger conversion channel enable bit (n=23-0,n≠19) 0: Disable 1: Enable | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.9 ADC conversion result rRegister (DATAx) x=23-0,x≠19

| Bit | Symbol | Description | Reset value |
|-------|--------|-----------------------|-------------|
| 31:12 | - | Reserved | - |
| 11:0 | RSLT | ADC conversion result | 0x0 |

17.5.10 ADC compare control register 0 (CMPx) x=0~1

| Bit | Symbol | Description | Reset value |
|-------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | ADCCMPxEN | ADC comparator x enable bit 0: - 1: Enable | 0 |
| 30 | ADCCMPxO | ADC comparator x result bit (read-only) (This bit is automatically updated after the selected channel is converted) 0: The conditions for comparison are not met 1: Comparison conditions are met | 0 |
| 29 | - | Reserved | - |
| 28 | ADCCMPxCOND | ADC comparator x compare condition select bit 0: ADC result < preset value 1: ADC result >= preset value | 0 |
| 27:24 | ADCCMPxMCNT | ADC comparator x match count preset value When the analog-to-digital conversion result of the specified channel matches the comparison condition, the internal counter will be incremented by 1, and when the internal counter equals to the value of ADCCMPxMCNT+1, the internal counter will be cleared to zero automatically. The internal counter will also be cleared to zero if the matching condition is not met during the accumulation process, i.e., this function has a filtering function. The ADC compare event is generated at the same time as the match, which can be used as a signal to trigger the brake operation of the EPWM. Note: The ADC Comparator 0 compare event will set the interrupt flag ADCCMP0IF to 1. | 0x0 |
| 23:21 | - | Reserved | - |
| 20:16 | ADCCMPxCHS | ADC comparator x compare channel select bit Note: 10011 disable selection 00000: Channel 0 10111: Channel 23 Other: Reserved | 0x0 |
| 15:12 | - | Reserved | - |
| 11:0 | ADCCMPxDATA | ADC comparator x data preset value (12-bit) | 0x0 |

17.5.11 ADC interrupt enable register (IMSC)

| Bit | Symbol | Description | Reset value |
|-------|-----------|------------------------------------------------------------------------------|-------------|
| 31 | ADCIMSC31 | ADC comparator 0 interrupt enable bit 0: Disable 1: Enable | 0 |
| 30:24 | - | Reserved | 0x0 |
| 23:0 | ADCIMSCn | ADC channel n interrupt enable bit (n=23-0, n≠19) 0: Disable 1: Enable | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.12 ADC interrupt source status register (RIS)

| Bit | Symbol | Description | Reset value |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | ADCRIS31 | ADC comparator 0 interrupt source status 0: No interrupt generated by interrupt sources 1: An interrupt is generated by interrupt sources | 0 |
| 30:24 | - | Reserved | 0x0 |
| 23:0 | ADCRISn | ADC channel n interrupt source status (n=23-0, n≠19) 0: No interrupt generated by interrupt sources 1: An interrupt is generated by interrupt sources | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.13 ADC enabled interrupt status register (MIS)

| Bit | Symbol | Description | Reset value |
|-------|----------|-------------------------------------------------------------------------------------------------------------------|-------------|
| 31 | ADCMIS31 | ADC comparator 0 interrupt status 0: No interrupt generated 1: Enable and generate an interrupt | 0 |
| 30:24 | - | Reserved | - |
| 23:0 | ADCMISn | ADC channel n interrupt status (n=23-0, n≠19) 0: No interrupt generated 1: Enable and generate an interrupt | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.14 ADC interrupt clear register (ICLR)

| Bit | Symbol | Description | Reset value |
|-------|-----------|-------------------------------------------------------------------------------------------------|-------------|
| 31 | ADCICLR31 | ADC comparator 0 interrupt status 1: Clear ADC Comparator 0 interrupt status 0: No effect | 0 |
| 30:24 | - | Reserved | - |
| 23:0 | ADCICLRn | ADC channel n interrupt status (n=23-0, n≠19) 0: No effect 1: Cleared | 0x0 |

Note: Bit19 is reserved and must be 0.

17.5.15 ADC write enable control register (LOCK)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0x55, enable the operation of ADC related registers. (For details, please refer to the description of ADC register mapping.) When LOCK=other values, disable the operation of ADC related registers. | 0x0 |

Chapter 18 Programmable Gain Amplifier (PGA0/1/2)

18.1 Overview

The chip contains three basic operational amplifier modules and three programmable gain amplifiers. Basic signal amplification and signal processing functions can be achieved with a few external components.

18.2 Features

PGA0 (Programmable Gain Amplifier 0)

- ◆ Adjustable gain: 1X/2X/2.5X/5X/7.5X/10X/15X.
- ◆ Positive input is selectable: from A0P input, or PGA_VREF (VREF/2 or BG).
- ◆ Supports pseudo-differential structure, feedback ground can be selected from the external port.
- ◆ Multiple output options for PGA0:
 - (1) Direct output to ADC channel 0, 1
 - (2) Direct output to comparator
 - (3) Direct output to PAD (A0O)
 - (4) Output to PAD (A0O) through a 10K resistor

PGA1 (Programmable Gain Amplifier 1)

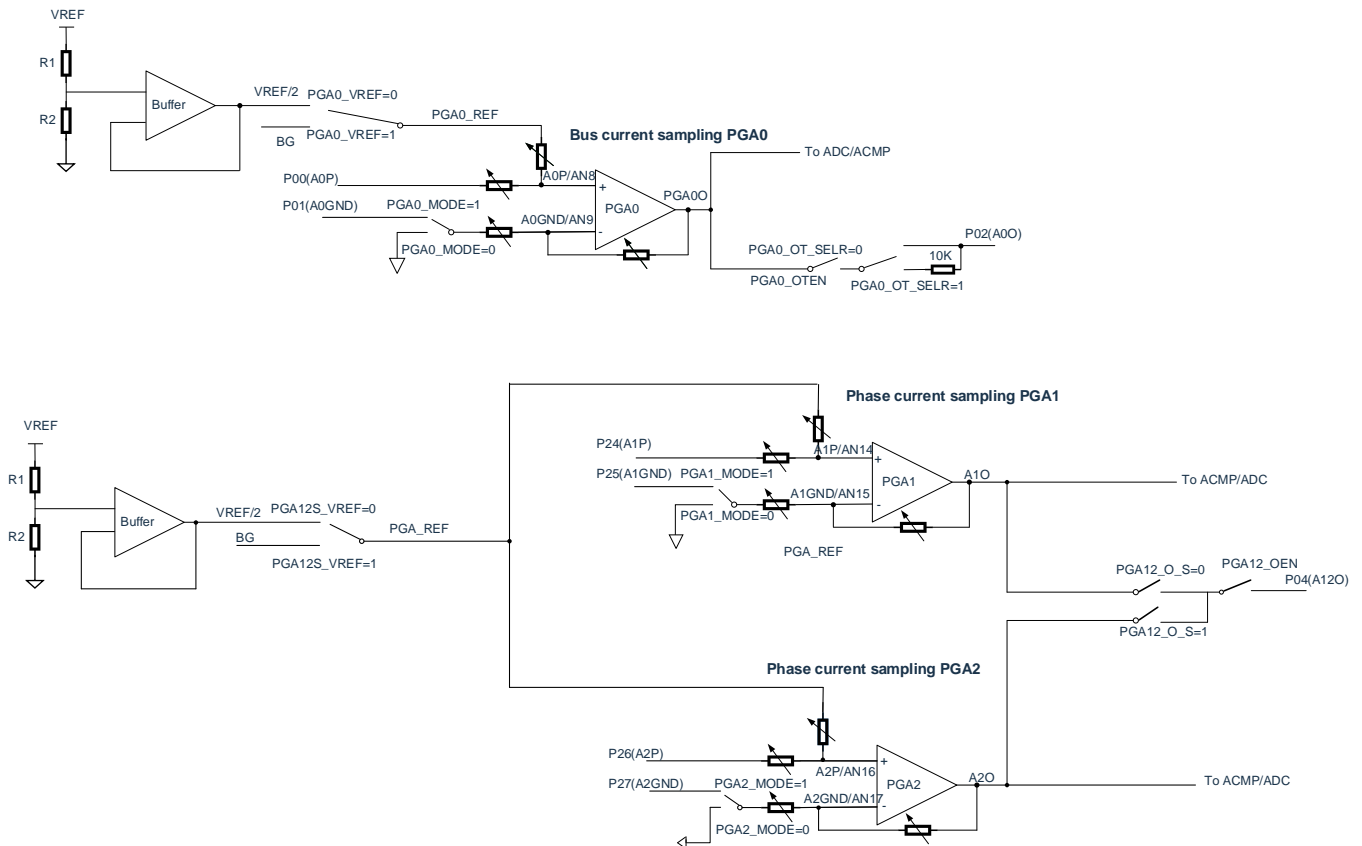
- ◆ Adjustable gain: 1X/2X/2.5X/5X/7.5X/10X/15X.
- ◆ Positive input is selectable: A1P input, or PGA_VREF(VREF/2 or BG).
- ◆ Supports pseudo-differential structure, feedback ground can be selected from the external port.
- ◆ Output options for PGA1:
 - (1) Direct output to ADC channel 2
 - (2) Direct output to comparator
 - (3) Direct output to PAD (A12O)

PGA2 (Programmable Gain Amplifier 2)

- ◆ Adjustable gain: 1X/2X/2.5X/5X/7.5X/10X/15X.
- ◆ Positive input is selectable: A2P input, PGA_VREF(VREF/2 or BG).
- ◆ Supports pseudo-differential structure, feedback ground can be selected from the external port.
- ◆ Output options for PGA2:
 - (1) Direct output to ADC channel 3
 - (2) Direct output to comparator
 - (3) Direct output to PAD (A12O)

18.3 Block diagram of structure

Figure 18-1: PGA structure diagram





18.4 Register mapping

(PGA0 base address = 0x4006_8300)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-----------------------------|-------------|
| PGA0CON0 | 0x000 | R/W | PGA0 Control Register 0 | 0x0 |
| PGA0CON1 | 0x004 | R/W | PGA0 Control Register 1 | 0x0 |
| PGA0LOCK | 0x008 | R/W | PGA0 Access Register Enable | 0x0 |

(PGA1/2 base address = 0x4006_8320)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|-----------|--------------|-----|------------------------------|-------------|
| PGA1CON0 | 0x000 | R/W | PGA1 Control Register | 0x0 |
| PGA2CON0 | 0x004 | R/W | PGA2 Control Register | 0x0 |
| PGA12CON | 0x008 | R/W | PGA12 Control Register | 0x0 |
| PGA12LOCK | 0x00C | R/W | PGA12 Access Register Enable | 0x0 |

18.5 Register description

18.5.1 PGA0 control register 0

| Bit | Symbol | Description | Reset value |
|------|------------|--------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7 | PGA0_EN | PGA0 enable bit 0: Disable 1: Enable | 0 |
| 6 | - | Reserved | - |
| 5 | PGA0_MODE | PG0 mode selection 0: Single-ended mode 1: Full differential mode | 0 |
| 4 | PGA0S_VREF | PGA0 reference voltage selection bit 0: VREF/2 1: BG (0.8V) | 0 |
| 3 | - | - | - |
| 2:0 | PGA0_S | PGA0 gain selection 000: 1X 001: 2X 010: 2.5X 011: 5X 100: 7.5X 101: 10X 110: 15X 111: 15X | 0x0 |

18.5.2 PGA0 control register 1

| Bit | Symbol | Description | Reset value |
|------|-------------|------------------------------------------------------------------------------------------------------------------|-------------|
| 31:4 | - | Reserved | - |
| 3 | PGA0_OTEN | PGA0 output to PAD channel enable 0: Disable 1: Enable | 0 |
| 2:1 | - | Reserved | - |
| 0 | PGA0_OT_SEL | PGA0 output to PAD series resistor selection 0: No internal resistors 1: With internal series 10K resistor | 0 |



18.5.3 PGA0 access enable register

| Bit | Symbol | Description | Reset value |
|------|-----------|----------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | PGA0_LOCK | PGA0 register access enable bit 0x55: Access to PGA0 related registers Other: Disable access | 0x0 |

18.5.4 PGA1 control register 0

| Bit | Symbol | Description | Reset value |
|------|-----------|--------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7 | PGA1_EN | PGA1 enable bit 0: Disable 1: Enable | 0 |
| 6:5 | - | Reserved | - |
| 4 | PGA1_MODE | PGA1 mode selection 0: Single-ended mode 1: Full differential mode | 0 |
| 3 | - | - | - |
| 2:0 | PGA1_S | PGA1 gain selection 000: 1X 001: 2X 010: 2.5X 011: 5X 100: 7.5X 101: 10X 110: 15X 111: 15X | 0x0 |



18.5.5 PGA2 control register 0

| Bit | Symbol | Description | Reset value |
|------|-----------|--------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7 | PGA2_EN | PGA2 enable bit 0: Disable 1: Enable | 0 |
| 6:5 | - | Reserved | - |
| 4 | PGA2_MODE | PGA2 mode selection 0: Single-ended mode 1: Full differential mode | 0 |
| 3 | - | - | - |
| 2:0 | PGA2_S | PGA2 gain selection 000: 1X 001: 2X 010: 2.5X 011: 5X 100: 7.5X 101: 10X 110: 15X 111: 15X | 0x0 |

18.5.6 PGA12 control register 0

| Bit | Symbol | Description | Reset value |
|-------|-------------|--------------------------------------------------------------------|-------------|
| 31:18 | - | Reserved | - |
| 17 | PGA12_OEN | PGA2/PGA1 output enable bit 0: Disable 1: Enable | 0 |
| 16 | PGA12_O_S | PGA2/PGA1 output select bit 0: PGA1 output 1: PGA2 output | 0 |
| 15:10 | - | - | - |
| 0 | PGA12S_VREF | PGA1/PGA2 reference voltage selection 0: VREF/2 1: BG (0.8V) | 0 |

18.5.7 PGA1/PGA2 access register enable

| Bit | Symbol | Description | Reset value |
|------|------------|-----------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | PGA12_LOCK | PGA1/PGA2 register access enable bit 0x55: Access PGA1/PGA2 related registers Other: Disable access | 0x0 |

Chapter 19 Analog Comparator (ACMP0/1)

19.1 Overview

The chip contains two analog comparators. The comparators can be configured to suit different applications. The comparators output a logic 1 when the positive voltage is greater than the negative voltage and a 0 when the negative voltage is greater than the positive voltage, which can also be changed via the output polarity selection bit. Each comparator can be configured to generate an interrupt when the comparator output value changes.

19.2 Block diagram of structure

Figure 19-1: Comparator block diagram

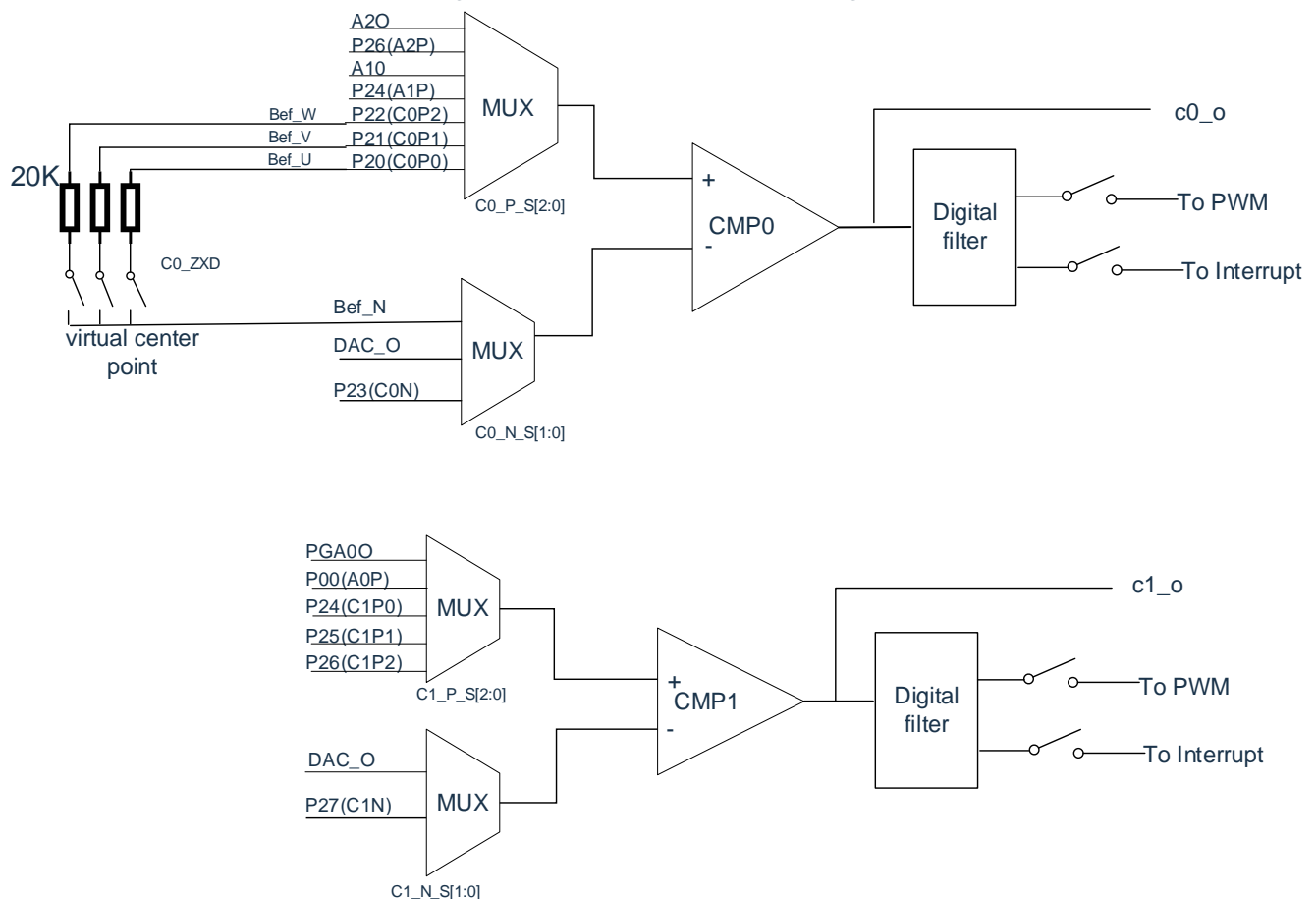
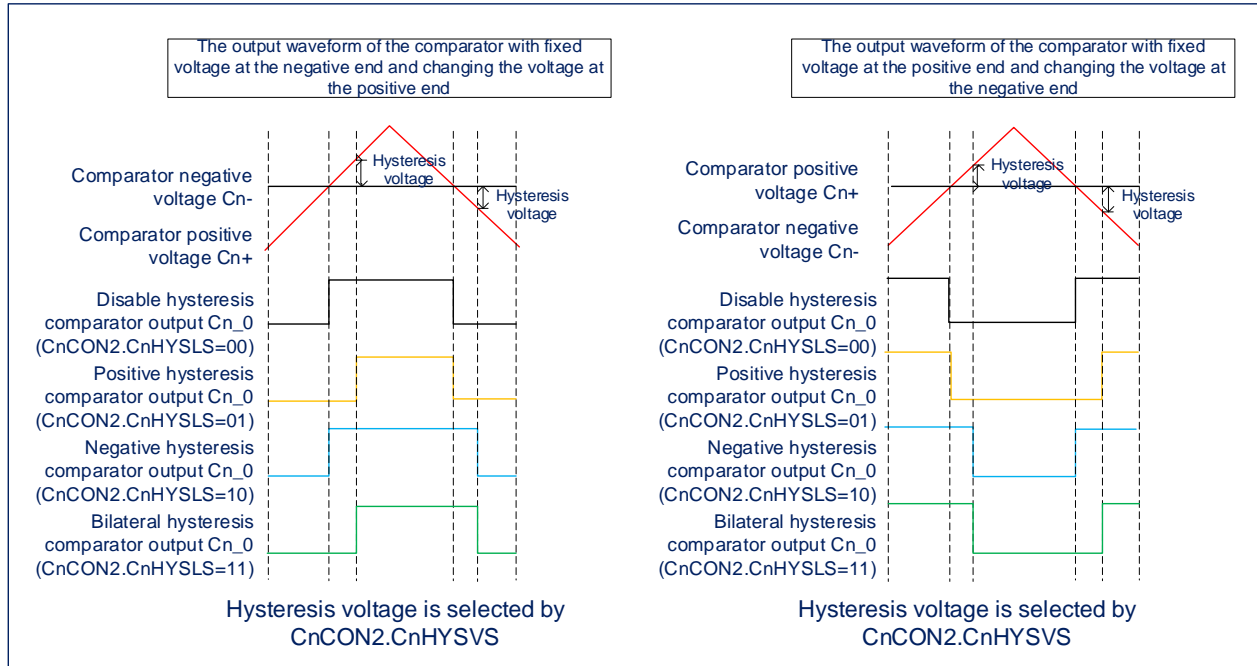


Figure 19-2: Block diagram of comparator hysteresis function

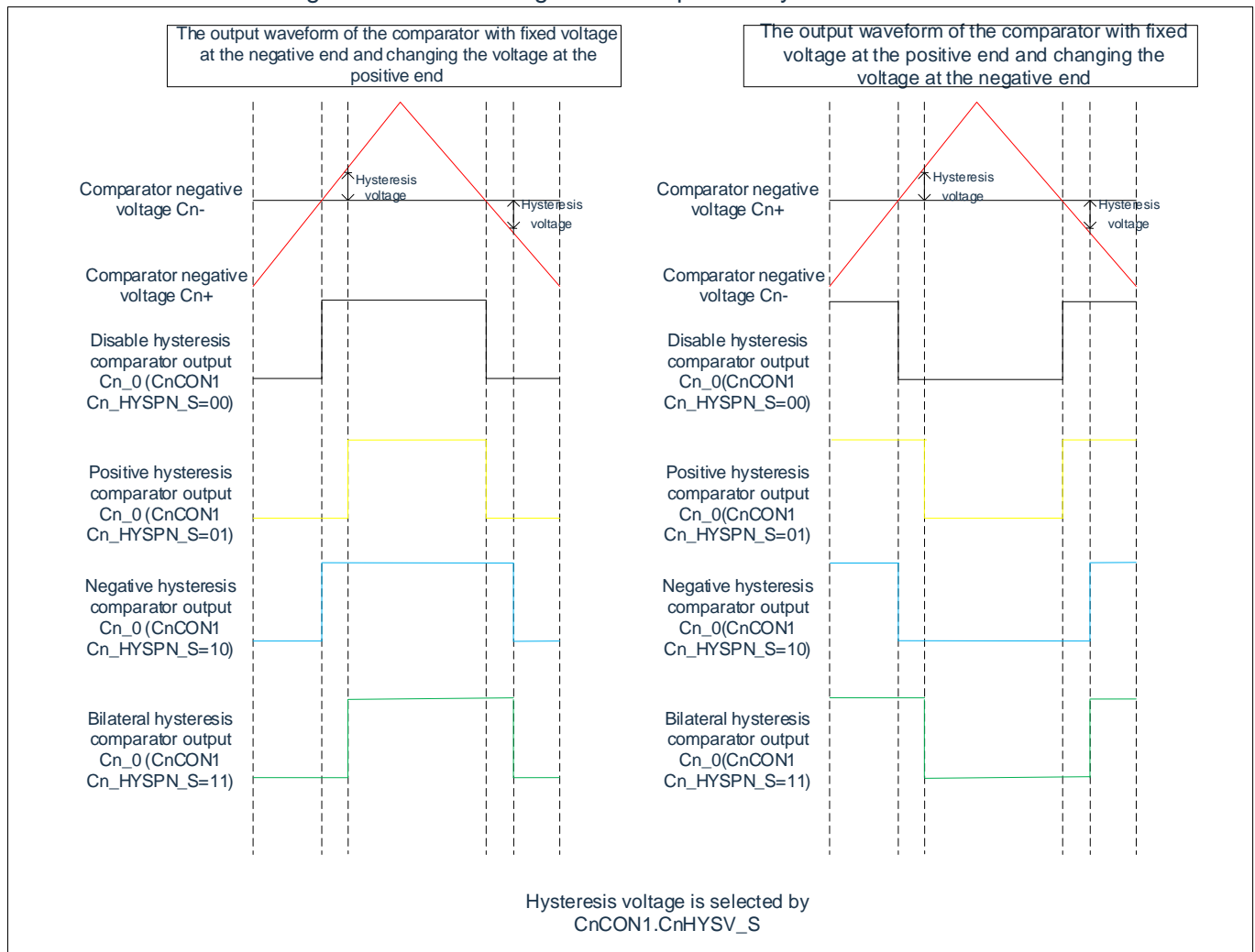


19.3 Features

- ◆ Analog input voltage range: (0~VDD)V.
- ◆ Supports single/bilateral hysteresis function.
- ◆ Supports hysteresis voltage selection (10mV/20mV/60mV – typical value).
- ◆ Each comparator's positive side can be selected from multiple sources.
- ◆ The negative side of each comparator is selectable between the port input and the internal reference voltage.
- ◆ Output filterable time selection: $0 \sim 512 \cdot T_{sys}$.
- ◆ Comparator event outputs can be used as brake trigger signals for enhanced PWM.
- ◆ Output change can generate an interrupt.

19.4 Function description

Figure 19-3: Block diagram of comparator hysteresis function



19.5 Register mapping

(ACMP base address = 0x4006_8200) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|-------------------------|--------------|-----|-----------------------------------------------------|-------------|
| C0CON0 _(P1B) | 0x000 | R/W | Analog Comparator 0 Control Register 0 | 0x0 |
| C0CON1 _(P1B) | 0x004 | R/W | Analog Comparator 0 Control Register 1 | 0x0 |
| C1CON0 _(P1B) | 0x008 | R/W | Analog Comparator 1 Control Register 0 | 0x0 |
| C1CON1 _(P1B) | 0x00C | R/W | Analog Comparator 1 Control Register 1 | 0x0 |
| CEVCON _(P1B) | 0x010 | R/W | Analog Comparator Event Control Register | 0x0 |
| IMSC _(P1B) | 0x014 | R/W | Analog Comparator Interrupt Enable Register | 0x0 |
| RIS | 0x018 | RO | Analog Comparator Interrupt Source Status Register | 0x0 |
| MIS | 0x01C | RO | Analog Comparator Enabled Interrupt Status Register | 0x0 |
| ICLR | 0x020 | WO | Analog Comparator Interrupt Clear Register | 0x0 |
| LOCK | 0x024 | R/W | Analog Comparator Write Enable Register | 0x0 |

Note:

The registers marked with (P1B) are protected registers.

When (P1B): LOCK==55H, the marked registers are allowed to be written; for other values, writing is prohibited

19.6 Register description

19.6.1 Analog comparator 0 control register 0 (C0CON0)

| Bit | Symbol | Description | Reset value |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:17 | - | Reserved | - |
| 16 | C0_ZXD | Analog comparator 0 center point select enable | |
| | | 0: Disable 1: Enable | |
| 15 | C0_EN | Analog comparator 0 enable bit 0: Disable 1: Enable | 0 |
| 14 | C0_OEN | Analog comparator 0 output enable bit 0: Disable 1: Enable | 0 |
| 13:9 | - | Reserved | - |
| 8 | C0_OUT | Analog comparator 0 result bit (read-only) | |
| 7 | - | Reserved | - |
| 6:4 | C0_P_S | Analog comparator 0 positive channel selection 000: C0P0 001: C0P1 010: C0P2 011: A1P 100: A1O 101: A2P 110: A2O 111: Disable | 0x0 |
| 3:2 | - | Reserved | - |
| 1:0 | C0_N_S | Analog comparator 0 negative channel selection 00: C0N 01: DAC_O 10: Bef_N virtual center point (comparator internal signal) 11: Disable | 0x0 |

19.6.2 Analog comparator 0 control register 1 (C0CON1)

| Bit | Symbol | Description | Reset value |
|-------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:14 | - | Reserved | - |
| 13:12 | C0_HYSPN_S | Analog comparator 0 hysteresis mode selection 00: No hysteresis 01: Positive hysteresis 10: Negative hysteresis 11: Positive and negative hysteresis | 0x0 |
| 11:10 | C0_HYSV_S | Analog comparator 0 hysteresis voltage selection 00: No hysteresis 01: 10mV 10: 20mV 11: 60mV | 0x0 |
| 9 | C0_POS | Analog comparator 0 output polarity select bit 0: Normal output 1: Inverted output | 0 |
| 8 | C0_FE | Analog comparator 0 output filter enable bit 0: Disable 1: Enable | 0 |
| 7:4 | - | Reserved | - |
| 3:0 | C0_FS | Analog comparator 0 output filter time select bit 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk Other: (0~1)*Tpclk | 0x0 |

19.6.3 Analog comparator 1 control register 0 (C1CON0)

| Bit | Symbol | Description | Reset value |
|-------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:16 | - | Reserved | - |
| 15 | C1_EN | Analog comparator 1 enable bit 0: Disable 1: Enable | 0 |
| 14 | C1_OEN | Analog comparator 1 output enable bit 0: Disable 1: Enable | 0 |
| 13:9 | - | Reserved - | - |
| 8 | C1_OUT | Analog comparator 1 result bit (read-only) | - |
| 7 | - | Reserved - | - |
| 6:4 | C1_P_S | Analog comparator 1 positive channel selection 000: C1P0 001: C1P1 010: C1P2 011: C1P3/A0P 100: PGA00 (PGA0 output) Other: Disable | 0x0 |
| 3:2 | - | Reserved | - |
| 1:0 | C1_N_S | Analog comparator 1 negative channel selection 00: C1N 01: DAC_O Other: Disable | 0x0 |

19.6.4 Analog comparator 1 control register 1 (C1CON1)

| Bit | Symbol | Description | Reset value |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:14 | - | Reserved | - |
| 13:12 | C1HYSPN_S | Analog comparator 1 hysteresis mode selection 00: No hysteresis 01: Positive hysteresis 10: Negative hysteresis 11: Positive and negative hysteresis | 0x0 |
| 11:10 | C1_HYSV_S | Analog comparator 1 hysteresis voltage selection 00: No hysteresis 01: 10mV 10: 20mV 11: 60mV | 0x0 |
| 9 | C1_POS | Analog comparator 1 output polarity select bit 0: Normal output 1: Inverted output | 0 |
| 8 | C1_FE | Analog comparator 1 output filter enable bit 0: Disable 1: Enable | 0 |
| 7:4 | - | Reserved | - |
| 3:0 | C1_FS | Analog comparator 1 output filter time select bit 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk Other: (0~1)*Tpclk | 0x0 |

19.6.5 Analog comparator event control register (CEVCON)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:6 | - | Reserved | - |
| 5 | EVE1 | Analog comparator 1 event output enable bit (does not affect interrupt generation) 0: Disable 1: Enable | 0 |
| 4 | EVE0 | Analog comparator 0 event output enable bit (does not affect interrupt generation) 0: Disable 1: Enable | 0 |
| 3:2 | EVS1 | Analog comparator 1 event generation condition select bit 00: Comparator 1 output jumps from 0->1 01: Comparator 1 output jumps from 1->0 10: Comparator 1 outputs a jump from 0->1 or a jump from 1->0 11: Reserved | 0x0 |
| 1:0 | EVS0 | Analog comparator 0 event generation condition select bit 00: Comparator 0 output jumps from 0->1 01: Comparator 0 output jumps from 1->0 10: Comparator 0 outputs a jump from 0->1 or a jump from 1->0 11: Reserved | 0x0 |

19.6.6 Analog comparator interrupt enable register (IMSC)

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------------------------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | EN_C1IF | Analog comparator 1 interrupt enable bit 0: Disable 1: Enable | 0 |
| 0 | EN_C0IF | Analog comparator 0 interrupt enable bit 0: Disable 1: Enable | 0 |

19.6.7 Analog comparator interrupt source status register (RIS)

| Bit | Symbol | Description | Reset value |
|------|----------|------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | RIS_C1IF | Analog comparator 1 interrupt source status bit 0: No interrupt generated 1: Interrupt has been generated (event | 0 |

| | | generation) | |
|---|----------|------------------------------------------------------------------------------------------------------------------------------------|---|
| 0 | RIS_C0IF | Analog comparator 0 interrupt source status bit 0: No interrupt generated 1: Interrupt has been generated (event generation) | 0 |

19.6.8 Analog comparator enabled interrupt source status register (MIS)

| Bit | Symbol | Description | Reset value |
|------|----------|---------------------------------------------------------------------------------------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | MIS_C1IF | Analog comparator 1 enabled interrupt status bit 0: No interrupt generated 1: An interrupt is generated | 0 |
| 0 | MIS_C0IF | Analog comparator 0 enabled interrupt status bit 0: No interrupt generated 1: An interrupt is generated | 0 |

19.6.9 Analog comparator interrupt clear control register (ICLR)

| Bit | Symbol | Description | Reset value |
|------|-----------|---------------------------------------------------------------------------------------------------|-------------|
| 31:2 | - | Reserved | - |
| 1 | ICLR_C1IF | Analog comparator 1 interrupt clear control bit 0: No effect 1: Clear the RIS_C1IF flag bit | 0 |
| 0 | ICLR_C0IF | Analog comparator 0 interrupt clear control bit 0: No effect 1: Clear the RIS_C0IF flag bit | 0 |

19.6.10 Analog comparator write enable control register (LOCK)

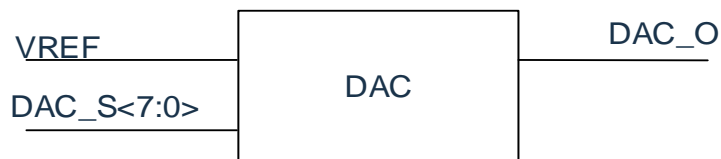
| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0x55, enable the operation of ACMP related registers. (For details, please refer to the description of ACMP register mapping.) When LOCK=other values, disable the operation of ACMP related registers. | 0x0 |

Chapter 20 DAC

20.1 Overview

The chip contains an internal digital-to-analog converter.

20.2 Block diagram of structure



20.3 Features

- ◆ The analog reference voltage input is the output of ADCLDO.
- ◆ Multiple levels of output voltage are available for selection.

20.4 Register mapping

(DAC base address = 0x4006_8360) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|---------------------------------|-------------|
| CON0 | 0x000 | R/W | DAC control register 0 | 0x0 |
| LOCK | 0x004 | R/W | DAC register enable control bit | 0x0 |

Note:

The registers marked with (P1B) are protected registers.

When (P1B): LOCK==55H, the marked registers are allowed to be written; for other values, writing is prohibited.

20.5 Register description

20.5.1 DAC control register 0(CON0)

| Bit | Symbol | Description | Reset value |
|------|--------|----------------------------------------------|-------------|
| 31:9 | - | Reserved | - |
| 8 | DAC_EN | DAC module enable 0: Disable 1: Enable | 0 |
| 7:0 | DAC_S | DAC digital signal input | 0x0 |

20.5.2 DAC write enable control register (LOCK)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0x55, enable the operation of DAC related registers. (For details, please refer to the description of ACMP register mapping.) When LOCK=other values, disable the operation of DAC related registers. | 0x0 |

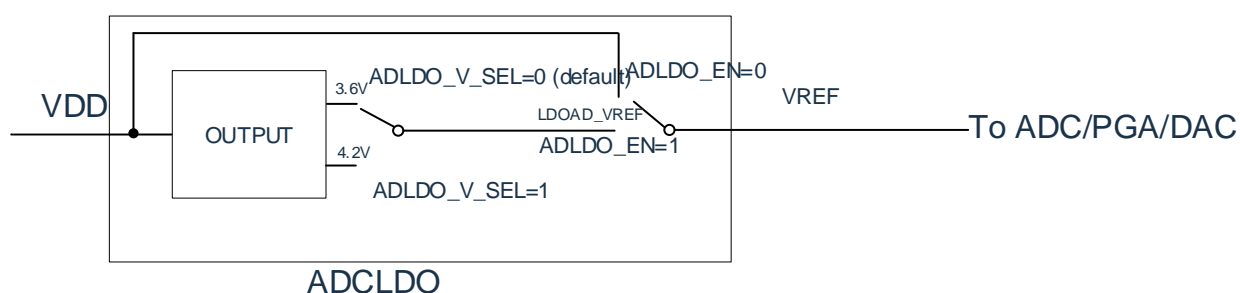
Chapter 21 Overview of ADCLDO

It contains an internal LDO to provide reference voltage to some modules.

21.1 Features

- ◆ Analog input voltage range: VDD.
- ◆ Output voltage: can be selected from VDD, 4.2V, and 3.6V.

21.2 Block diagram of structure



21.3 Register mapping

(ADCLDO base address = 0x4006_8340) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------------|-------------|
| CON0 | 0x000 | R/W | ADCLDO control register 0 | 0x0 |
| LOCK | 0x00C | R/W | ADCLDO register enable control bit | 0x0 |

Note: When LOCK == 55H, CON0 is allowed to be written; = other values, it is forbidden to be written.

21.4 Register description

21.4.1 ADCLDO control register 0(CON0)

| Bit | Symbol | Description | Reset value |
|-------|----------------|----------------------------------------------------------------------------------------------------|-------------|
| 31:21 | - | Reserved | - |
| 20:16 | ADCLDO_AJ[4:0] | ADCLDO trim bit (read-only) | |
| 15:9 | - | Reserved | 0x0 |
| 8 | ADCLDO_EN | ADCLDO module enable | 0 |
| | | 0: Disable, ADLDO output VDD 1: Enable, ADLDO output LDO voltage | |
| 7:0 | ADCLDO_V_SEL | ADCLDO output voltage selection 0x55: LDO voltage output 4.2V Other: LDO voltage output 3.6V | 0x0 |

21.4.2 ADCLDO write enable control register (LOCK)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | LOCK | When LOCK=0x55, enable the operation of ADCLDO related registers. (For details, please refer to the description of ACMP register mapping.) When LOCK=other values, disable the operation of ADCLDO related registers. | 0x0 |



Chapter 22 Nested Vectored Interrupt Controller (NVIC)

Cortex®-M0+ CPU includes a Nested Vectored Interrupt Controller (NVIC) for interrupt handling.

22.1 Features

- ◆ Supports nested vectored interrupts.
- ◆ Automatically saves and restores processor state.
- ◆ Allows dynamic priority changes.
- ◆ Simplified and defined interrupt times.

The NVIC processes all supported exceptions based on their priority. All exceptions are handled in “Handler mode”. The NVIC supports 21 discrete interrupts (IRQ[31:0]), with each interrupt supporting 4 levels of interrupt priority. All interrupts and most system exceptions can be configured to have different priorities. When an interrupt occurs, the NVIC compares the priority of the new interrupt with the current interrupt. If the new interrupt has a higher priority, it is immediately processed.

After accepting an interrupt, the start address of the Interrupt Service Routine (ISR) can be obtained from the vector table in memory. Software does not need to determine which interrupt is being responded to or allocate the start address of the relevant ISR. Once the start address is obtained, the NVIC automatically saves the values of processor state registers (PC, PSR, LR, R0~R3, R12) to the stack. After the ISR ends, the NVIC restores the values of the relevant registers from the stack and resumes normal operation. This allows for minimal and identified interrupt handling time.

The NVIC supports “Tail-Chaining”, which efficiently handles back-to-back interrupts without saving and restoring the current state, reducing the latency for switching from the current ISR to a pending ISR. The NVIC also supports “Late Arrival”, improving the efficiency of concurrent interrupts. When a higher-priority interrupt request occurs before the current ISR begins executing (during the stage of saving processor state and obtaining the start address), the NVIC immediately processes the higher-priority interrupt, thus enhancing real-time performance.

For more detailed information, please refer to the “ARM® Cortex®-M0+ Technical Reference Manual” and the “ARM®v6-M Architecture Reference Manual”.

22.2 Exception mode and system interrupt mapping

The table below lists the exception modes supported by this product. Like all interrupts, software can set 4 levels of priority for some of these exceptions. Users can configure the highest priority as 0 and the lowest priority as 3. The default priority for all user-configurable interrupts is 0.

| Exception Name | Exception Number | Priority |
|------------------------|------------------|--------------|
| Reset | 1 | -3 |
| NMI | 2 | -2 |
| Hard Fault | 3 | -1 |
| Reserved | 4~10 | Reserved |
| SVCall | 11 | Configurable |
| Reserved | 12~13 | Reserved |
| PendSV | 14 | Configurable |
| SysTick | 15 | Configurable |
| Interrupt (IRQ0~IRQ31) | 16~47 | Configurable |

Note: Priority 0 is the 4th priority in the system, after “Reset”, “NMI” and “Hard Fault”.

22.3 Vector table

| Exception number | Interrupt number | Vector address | Exception type | Description |
|------------------|------------------|----------------|------------------|---------------------------------------------------------------------------|
| 1-15 | - | 0x00-0x3c | System exception | - |
| 16 | 0 | 0x40 | INTLVI | Voltage detection |
| 17 | 1 | 0x44 | INTP0 | Edge detection of pin input |
| 18 | 2 | 0x48 | INTP1 | Edge detection of pin input |
| 19 | 3 | 0x4c | INTP2 | Edge detection of pin input |
| 20 | 4 | 0x50 | INTP3 | Edge detection of pin input |
| 21 | 5 | 0x54 | INTTM01H | Timer channel 01 counting or capture end (high 8 bits of timer operation) |
| 22 | 6 | 0x58 | INTCCP | CCP interrupt |
| 23 | 7 | 0x5c | INTEPWM | EPWM interrupt |
| 24 | 8 | 0x60 | INTADC | ADC interrupt |
| 25 | 9 | 0x64 | INTACMP | Comparator completion interrupt |
| 26 | 10 | 0x68 | INTUART0 | UART0 interrupt |
| 27 | 11 | 0x6c | INTI2C | IIC interrupt |
| 28 | 12 | 0x70 | INTSPI | SPI interrupt |
| 29 | 13 | 0x74 | INTTIMER0 | TIMER0 interrupt |
| 30 | 14 | 0x78 | INTTIMER1 | TIMER1 interrupt |
| 31 | 15 | 0x7c | INTLSITIMER | LSI timer interrupt |
| 32 | 16 | 0x80 | Reserved | - |
| 33 | 17 | 0x84 | INTTM00 | Timer channel 00 count end or capture end |
| 34 | 18 | 0x88 | INTTM01 | Timer channel 01 count end or capture end |
| 35 | 19 | 0x8c | INTTM02 | Timer channel 02 count end or capture end |
| 36 | 20 | 0x90 | INTTM03 | Timer channel 03 count end or capture end |
| 37 | 21 | 0x94 | Reserved | - |
| 38 | 22 | 0x98 | Reserved | - |
| 39 | 23 | 0x9c | Reserved | - |
| 40 | 24 | 0xa0 | Reserved | - |
| 41 | 25 | 0xa4 | Reserved | - |
| 42 | 26 | 0xa8 | Reserved | - |
| 43 | 27 | 0xac | Reserved | - |
| 44 | 28 | 0xb0 | Reserved | - |
| 45 | 29 | 0xb4 | Reserved | - |
| 46 | 30 | 0xb8 | Reserved | - |
| 47 | 31 | 0xbc | INTFL | FLASH programming completed |



22.4 Register mapping

(NVIC base address = 0xE000_E000) RO: Read only, WO: Write Only, R/W: Read/Write.

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|------------------------------------------|-------------|
| ISER | 0x100 | R/W | Interrupt Enable Control Register | 0x0 |
| ICER | 0x180 | R/W | Interrupt Clear Enable Control Register | 0x0 |
| ISPR | 0x200 | R/W | Interrupt Set Pending Control Register | 0x0 |
| ICPR | 0x280 | R/W | Interrupt Clear Pending Control Register | 0x0 |
| IPR0 | 0x400 | R/W | IRQ0~IRQ3 Interrupt Priority Register | 0x0 |
| IPR1 | 0x404 | R/W | IRQ4~IRQ7 Interrupt Priority Register | 0x0 |
| IPR2 | 0x408 | R/W | IRQ8~IRQ11 Interrupt Priority Register | 0x0 |
| IPR3 | 0x40C | R/W | IRQ12~IRQ15 Interrupt Priority Register | 0x0 |
| IPR4 | 0x410 | R/W | IRQ16~IRQ19 Interrupt Priority Register | 0x0 |
| IPR5 | 0x414 | R/W | IRQ20~IRQ23 Interrupt Priority Register | 0x0 |
| IPR6 | 0x418 | R/W | IRQ24~IRQ27 Interrupt Priority Register | 0x0 |
| IPR7 | 0x41C | R/W | IRQ28~IRQ31 Interrupt Priority Register | 0x0 |

(INTM base address = 0x4004_5B38) RO: Read only, WO: Write Only, R/W: Read/Write.

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-------------------------------------------------|-------------|
| EGP0 | 0x000 | R/W | External Interrupt Rising Edge Enable Register | 0x0 |
| EGN0 | 0x001 | R/W | External Interrupt Falling Edge Enable Register | 0x0 |

22.5 Register description

22.5.1 Interrupt set enable control register (ISER)

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:0 | SETENA | <p>Interrupt enable bit Enables one or more interrupts. Each bit represents an interrupt from IRQ0 to IRQ31 (vector number from 16 to 47). Write operation:</p> <ul style="list-style-type: none"> 0: Invalid 1: Enables corresponding interrupt(s) <p>Read operation:</p> <ul style="list-style-type: none"> 0: Disable state of the corresponding interrupt(s) 1: Enable state of the corresponding interrupt(s) <p>Note: Reading the value of this register indicates that it is currently enabled.</p> | 0x0 |

22.5.2 Interrupt clear enable control register (ICER)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:0 | CLRENA | <p>Interrupt disable bit Disable one or more interrupts. Each bit represents an interrupt from IRQ0 to IRQ31 (vector number from 16 to 47). Read operation:</p> <ul style="list-style-type: none"> 0: Invalid 1: Related interrupts are prohibited <p>Read operation:</p> <ul style="list-style-type: none"> 0: Disable related interrupt status 1: Enable related interrupt status <p>Note: Reading the value of this register indicates that it is currently enabled.</p> | 0x0 |

22.5.3 Interrupt set pending control register (ISPR)

| Bit | Symbol | Description | Reset value |
|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:0 | SETPEND | <p>Set interrupt pending bit Write operation:</p> <ul style="list-style-type: none"> 0: Invalid 1: Sets the pending state. Each bit represents an interrupt from IRQ0 to IRQ31 (vector number from 16 to 47). <p>Read operation:</p> <ul style="list-style-type: none"> 0: The relevant interrupt is not pending 1: The relevant interrupt is in the | 0x0 |



| | | | |
|--|--|-------------------------------------------------------------------------------------------|--|
| | | pending state Note: Reading this register indicates that the current state is pending. | |
|--|--|-------------------------------------------------------------------------------------------|--|

22.5.4 Interrupt clear pending control register (ICPR)

| Bit | Symbol | Description | Reset value |
|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:0 | CLRPEND | <p>Clear interrupt pending bit Write operation:</p> <ul style="list-style-type: none"> 0: Invalid 1: Clear pending status. Each bit represents an interrupt from IRQ0 to IRQ31 (vector number from 16 to 47). <p>Read operation:</p> <ul style="list-style-type: none"> 0: The relevant interrupt is not pending 1: The relevant interrupt is in the pending state <p>Note: Reading this register indicates that the current state is pending.</p> | 0x0 |

22.5.5 IRQ0~IRQ3 Interrupt priority register (IPR0)

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_3 | IRQ3 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_2 | IRQ2 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_1 | IRQ1 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_0 | IRQ0 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.6 IRQ4~IRQ7 Interrupt priority register (IPR1)

| Bit | Symbol | Description | Reset value |
|-------|--------|-------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_7 | IRQ7 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_6 | IRQ6 priority | 0x0 |



| | | | |
|-------|-------|-------------------------------------------------------------------------------------------|-----|
| | | 0 represents the highest priority, and 3 represents the lowest priority. | |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_5 | IRQ5 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_4 | IRQ4 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.7 IRQ8~IRQ11 Interrupt priority register (IPR2)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_11 | IRQ11 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_10 | IRQ10 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_9 | IRQ9 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_8 | IRQ8 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.8 IRQ12~IRQ15 Interrupt priority register (IPR3)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_15 | IRQ15 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_14 | IRQ14 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_13 | IRQ13 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_12 | IRQ12 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.9 IRQ16~IRQ19 Interrupt priority register (IPR4)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_19 | IRQ19 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_18 | IRQ18 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_17 | IRQ17 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_16 | IRQ16 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.10 IRQ20~IRQ23 Interrupt priority register (IPR5)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_23 | IRQ23 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_22 | IRQ22 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_21 | IRQ21 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_20 | IRQ20 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.11 IRQ24~IRQ27 Interrupt priority register (IPR6)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_27 | IRQ27 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_26 | IRQ26 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_25 | IRQ25 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_24 | IRQ24 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.12 IRQ28~IRQ31 Interrupt priority register (IPR7)

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:30 | PRI_31 | IRQ31 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 29:24 | - | Reserved | - |
| 23:22 | PRI_30 | IRQ30 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 21:16 | - | Reserved | - |
| 15:14 | PRI_29 | IRQ29 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 13:8 | - | Reserved | - |
| 7:6 | PRI_28 | IRQ28 priority 0 represents the highest priority, and 3 represents the lowest priority. | 0x0 |
| 5:0 | - | Reserved | - |

22.5.13 External interrupt rising edge enable register (EGP0)

EGP0 and EGN0 registers are used to set the active edge of INTP0~INTP3. The EGP0 and EGN0 registers are set by 8-bit memory operation instructions.

After a reset signal is generated, the values of these registers are changed to “00H”.

Figure 22-1: Format of external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0)

| Bit | Symbol | Description | Reset value |
|-----|--------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:4 | - | Reserved | 0x0 |
| 3 | EGP3 | INTP3 external interrupt rising edge enable register: 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge | 0 |
| 2 | EGP2 | INTP2 external interrupt rising edge enable register: 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge | 0 |
| 1 | EGP1 | INTP1 external interrupt rising edge enable register: 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge | 0 |
| 0 | EGP0 | INTP0 external interrupt rising edge enable register: 0: Disable external interrupt rising edge 1: Enable external interrupt rising edge | 0 |

22.5.14 External interrupt falling edge enable register (EGN0)

| Bit | Symbol | Description | Reset value |
|-----|--------|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:4 | - | Reserved | 0x0 |
| 3 | EGN3 | INTP3 external interrupt falling edge enable register: 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge | 0 |
| 2 | EGN | INTP2 external interrupt falling edge enable register: 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge | 0 |
| 1 | EGN | INTP1 external interrupt falling edge enable register: 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge | 0 |
| 0 | EGN | INTP0 external interrupt falling edge enable register: 0: Disable external interrupt falling edge 1: Enable external interrupt falling edge | 0 |

| EGPn | EGNn | Effective edge selection on INTPn pin (n=0~3) |
|------|------|-----------------------------------------------|
| 0 | 0 | Disable detection of edges. |
| 0 | 1 | Falling edge |
| 1 | 0 | Rising edge |
| 1 | 1 | Both edges |

The ports corresponding to the EGPn and EGNn bits are shown in Table 22-1.

Table 22-1: Interrupt request signal corresponding to EGPn bit and EGNn bit

| Detect enable bit | | Interrupt request signal |
|-------------------|------|--------------------------|
| EGP0 | EGN0 | INTP0 |
| EGP1 | EGN1 | INTP1 |
| EGP2 | EGN2 | INTP2 |
| EGP3 | EGN3 | INTP3 |

Note 1: If the input port used for the external interrupt function is switched to output mode, an active edge may be detected and an INTPn interrupt may be generated. When switching to the output mode, the port mode register (PMxx) must be set to “0” after disabling the detection of an edge (EGPn, EGNn=0, 0).

Note 2: Please refer to “3.1 Port Function” for the ports with edge detection.

Note 3: n=0~3

Chapter 23 Standby Function

23.1 Standby function

The standby function is a function that further reduces the operating current of the system and has the following two modes.

(1) Sleep mode

Sleep mode is the mode in which the CPU is stopped from running the clock. If the high-speed on-chip oscillator or the low-speed on-chip oscillator is oscillating before the sleep mode is set, the clocks continue to oscillate. Although this mode does not reduce the operating current to the level of deep sleep mode, it is an effective mode for wanting to restart processing immediately through interrupt requests or if you want to run frequently in intermittent operations.

(2) Deep sleep mode

Deep sleep mode is a mode that stops the oscillation of the high-speed on-chip oscillator and stops the entire system. The operating current of the CPU can be greatly reduced.

Because deep sleep mode can be released by interrupt requests, intermittent operations can also be performed. However, because the wait time to ensure oscillation stability is required when releasing the deep sleep mode, it is necessary to select the sleep mode if you need to start processing immediately through the interrupt request.

In either mode, registers, flags, and data memory are all left set to before standby mode, and the output latches and output buffers of the input/output ports are also maintained.

Note 1: When shifting to the deep sleep mode, WFI instructions must be executed after stopping peripheral hardware running in the master system clock.

Note 2: To reduce the operating current of the A/D converter, after clearing bit 4 (ADCEN) of the A/D converter control register (CON) and bit 7 (The ADCST conversion is completed by clearing 0 via hardware.) of the control register (CON2) to "0", execute the WFI instruction after stopping the A/D conversion operation.

Note 3: The option byte can be used to select whether to continue or stop the low-speed internal oscillator in sleep mode or deep sleep mode. For details, please refer to "Chapter 29 Option Byte".

23.2 Sleep mode

23.2.1 Setting of sleep mode

When the SLEEPDEEP bit of the SCR register is 0, execute the WFI instruction and enter sleep mode. In sleep mode, the CPU stops operating, but the values of the internal registers are still maintained, and the peripheral modules remain in the state they were in before they entered sleep mode. The status of peripheral modules, oscillators, etc. in sleep mode is shown in Table 23-1.

Sleep mode can be set regardless of whether the CPU clock before setup is a high-speed on-chip oscillator clock or a low-speed on-chip oscillator clock.

Caution: When the interrupt request flag is “1” (an interrupt request signal is generated), the interrupt request signal is used to release the sleep mode. Therefore, even if the WFI instruction is executed in this case, it does not shift to the sleep mode.

Table 23-1: Operation status in sleep mode

| Item | | Sleep mode |
|-----------------------------------------------------------------|---------------------|------------------------------------------------------------------------------------------------------------|
| System clock | F _{IH} | Operation continues |
| | F _{IL} | The operating state is set via the OSMC register and the SUBCKSEL register, and the set state is retained. |
| CPU | | Operation stopped |
| Code flash memory | | Operation stopped |
| RAM | | Operation stopped |
| Port (latch) | | Status before sleep mode was set is retained |
| General-purpose timer unit TIMER4 | | Operable |
| LSI_Timer | | Operable |
| Clock output/buzzer output | | Operable |
| Watchdog timer | | If counting is set to continue before sleep, counting can continue after sleep. |
| DIVSQRT Unit | | Calculation stopped |
| TIMER0/1 | | Operable |
| Capture/Compare/PWM (CCP0/1) | | Operable |
| Enhanced PWM(EPWM) | | Operable |
| UART | | Operable |
| I ² C Serial Interface Controller (I ² C) | | Operable |
| Serial Peripheral Interface Controller (SSP/SPI) | | Operable |
| Analog-to-digital conversion (ADC) | | Operable |
| Programmable Gain Amplifier (PGA0/1/2) | | Operable |
| Analog comparator (ACMP0/1) | | Operable |
| DAC | | Operable |
| ADCLDO | | Operable |
| Power-on reset function | | Operable |
| Voltage detection function | | Operable |
| External interrupt | | Operable |
| CRC | High-speed CRC | Operable |
| | General-purpose CRC | Operation stopped |
| SFR guard function | | Operation stopped |

Note: Operation stopped: Operation is automatically stopped before switching to the sleep mode.

F_{IH}: High-speed on-chip oscillator clock

F_{IL}: Low-speed on-chip oscillator clock

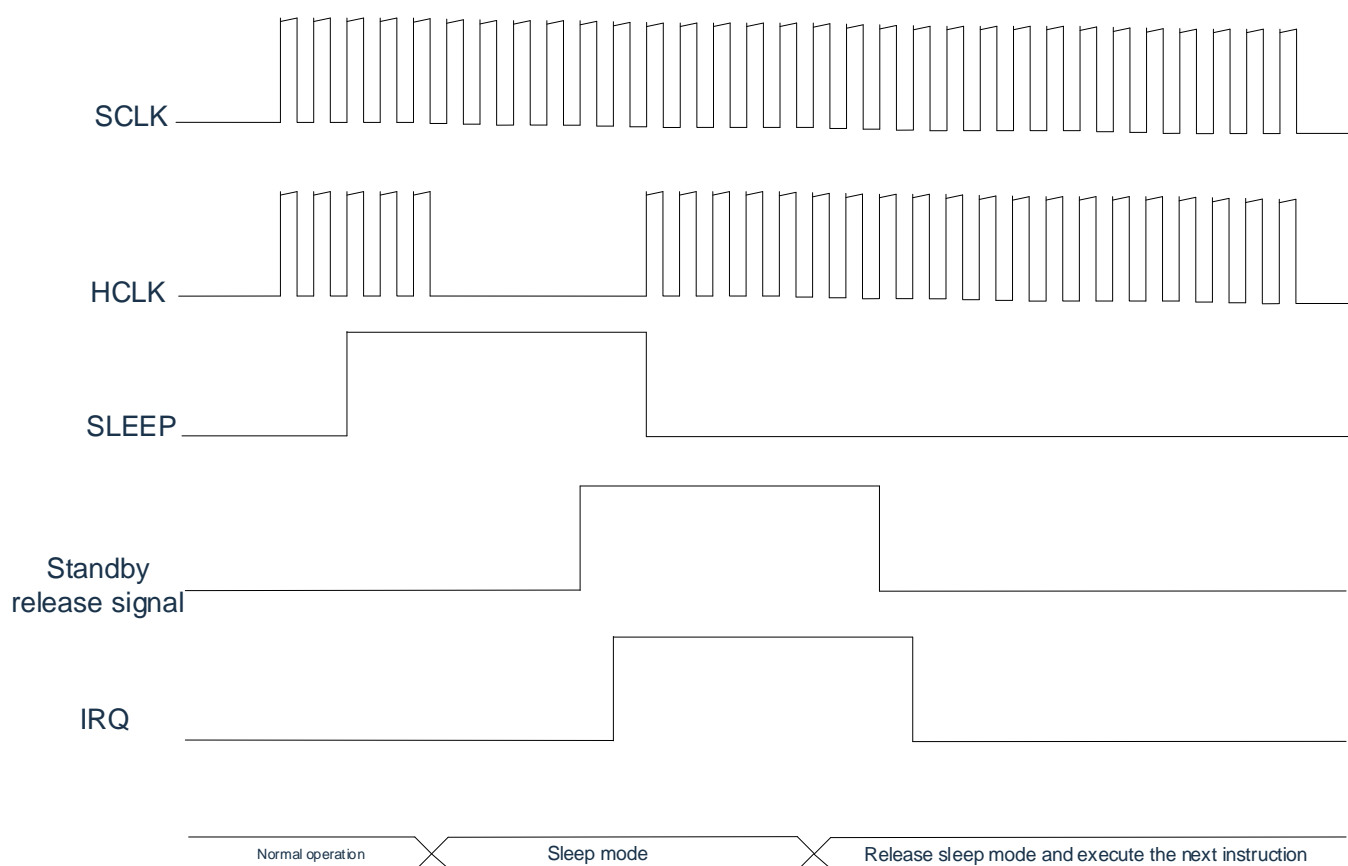
23.2.2 Sleep mode release

The sleep mode can be released by any interrupt or external reset, POR reset, low voltage detection reset, or WDT reset.

(1) Released by interrupts

When a interrupt is generated and the interrupt is allowed to be accepted, sleep mode is released and the CPU begins processing interrupt services.

Figure 23-1: Release sleep mode by interrupt requests



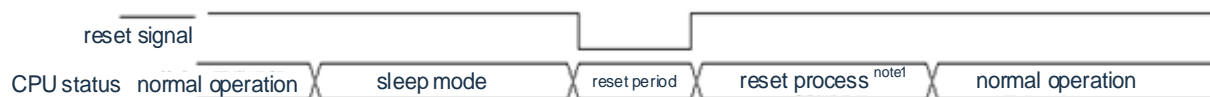
Note 1: From the generation of the standby release signal to the release of sleep mode, it takes 16 clocks to start executing the interrupt service program.

Note 2: Some of the standby release signals cannot be cleared by themselves, but must be cleared by writing to a register. This is usually done by writing to a register in the interrupt service program.

(2) Released by resets

When a reset signal is generated, the CPU is in reset state and the sleep mode is released. As with a normal reset, the program is executed after shifting to the reset vector address.

Figure 23-2: Release sleep mode by resets



Note: For reset processing, please refer to “Chapter 24 Reset Function”. For reset processing of power-on reset (POR) and voltage detection (LVD), refer to “Chapter 25 Power-on Reset Circuit”.

23.3 Deep sleep mode

23.3.1 Setting of deep sleep mode

When the SLEEPDEEP bit of the SCR register is 1, the WFI instruction is executed and deep sleep mode is entered. In this mode, the CPU, most of the peripheral modules, and the oscillator operation stops. However, the values of the CPU internal registers, the RAM data, the peripheral modules, the state of the I/O are maintained. The operating status of the peripheral module and the oscillator in deep sleep mode is shown in Table 23-2.

Note: When the interrupt request flag is “1” (an interrupt request signal is generated), the interrupt request signal is used to release deep sleep mode. Therefore, if the WFI instruction is executed in this case, it is released as soon as it enters deep sleep mode. Returns to operation mode after executing the WFI instruction and after a deep sleep mode release time has elapsed.

Table 23-2: Operation status in deep sleep mode

| Item | | Deep sleep mode |
|-----------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------|
| System clock | F _{IH} | Operation stopped |
| | F _{IL} | The operating state is set via the OSMC register and the SUBCKSEL register, and the set state is retained. |
| CPU | | Operation stopped |
| Code flash memory | | Operation stopped |
| RAM | | Operation stopped |
| Port (latch) | | Status before deep sleep mode was set is retained |
| General-purpose timer unit TIMER4 | | Operation disabled |
| LSI_Timer | | Operable |
| Clock output/buzzer output | | Operation disabled |
| Watchdog timer | | Operable |
| DIVSQRT Unit | | Calculation stopped |
| TIMER0/1 | | Operation disabled |
| Capture/Compare/PWM (CCP0/1) | | Operation disabled |
| Enhanced PWM(EPWM) | | Operation disabled |
| UART | | Operation disabled |
| I ² C Serial Interface Controller (I ² C) | | Operation disabled |
| Serial Peripheral Interface Controller (SSP/SPI) | | Operation disabled |
| Analog-to-digital conversion (ADC) | | Operation disabled |
| Programmable Gain Amplifier (PGA0/1/2) | | Operation disabled |
| Analog comparator (ACMP0/1) | | Operation disabled |
| DAC | | Operation disabled |
| ADCLDO | | Operation disabled |
| Power-on reset function | | Operable |
| Voltage detection function | | Operable |

| | | |
|--------------------|---------------------|-------------------|
| External interrupt | | Operable |
| CRC | High-speed CRC | Operation stopped |
| | General-purpose CRC | Operation stopped |
| SFR guard function | | Operation stopped |

Note: Operation stopped: Operation is automatically stopped before switching to the deep sleep mode.

Operation disabled: Operation is stopped before switching to the deep sleep mode.

F_{IH} : High-speed on-chip oscillator F_{IL} : Low-speed on-chip oscillator

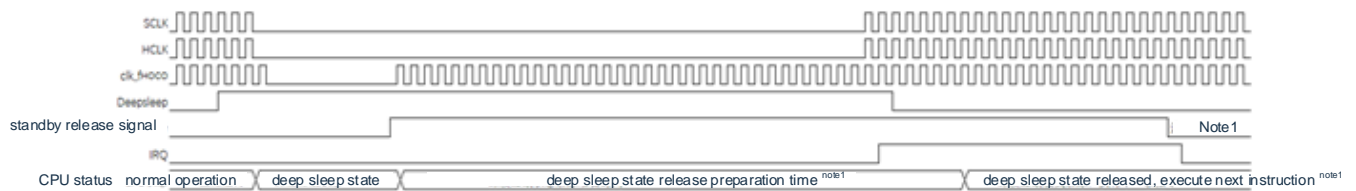
23.3.2 Deep sleep mode release

The deep sleep mode can be released by the following two methods.

(a) Released by non-maskable interrupt requests

If an LVD detection, INTP0-3, LSI timer or WDT interrupt request occurs, the deep sleep mode is released. After the oscillation stabilization time, if it is allowed to accept interrupt, it will process the vector interrupt. If it is not allowed to accept interrupt, it executes the instruction at the next address.

Figure 23-3: Release deep sleep mode by interrupt requests



Note 1: Standby release signal: For details of the standby release signal, refer to the section on Interrupt.

Note 2: Deep sleep release preparation time:

When the CPU clock is a high-speed on-chip oscillation clock before entering deep sleep mode: at least 20us.

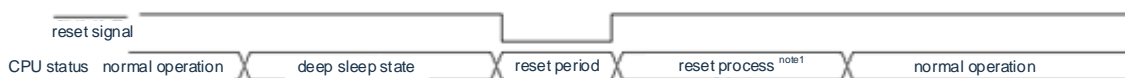
Note 3: Wait: 14 clocks are required from when the time CPU.IRQ is valid to the interrupt service program starts.

Note 4: The oscillation accuracy of the high-speed on-chip oscillator clock varies steadily depending on temperature conditions and during deep sleep mode.

(b) Released by generating reset signals

The deep sleep mode is released by generating a reset signal. Then, as with a normal reset, the program is executed after shifting to the reset vector address.

Figure 23-4: Release deep sleep mode by resetting



Note: For reset processing, see “Chapter 24 Reset Function”. For reset processing of power-on reset (POR) and voltage detection (LVD), see “Chapter 25 Power-on Reset Circuit”.

23.4 Deep sleep mode with partial power down

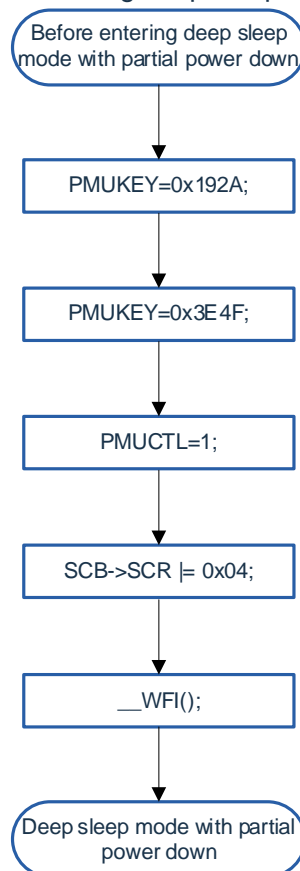
23.4.1 Setting of deep sleep mode with partial power down

The deep sleep mode with partial power loss is a deep sleep mode that further saves power consumption by turning off some peripheral power supplies on the basis of deep sleep mode. Enter the partial power-down deep sleep mode needs to configure the PWDNEEN bit of the PMUCTL register, the control bit is written to the power supply mode control protection register (PMUKEY) protection, when the deep sleep mode of partial power down requires reinitialization of the power-down peripheral before it can re-operate normally, please refer to Table 23-3 The operation status in the deep sleep mode of the partial power-down is required for details.

When the SCR register has a SLEEPDEEP bit of 1 and the PMUCTL register has a PWDNEEN bit of 1, executing the WFI command can enter a partially powered-down deep sleep mode. In this mode, the CPU and the transmitter stop functioning, and most peripheral modules are powered off. However, the value of the CPU's internal registers, RAM data, the state of the I/O is maintained. The operating status of the peripheral module and the oscillator in the deep sleep mode of partial power failure is shown in Table 23-3.

The PWDNEEN bit of the PMUCTL register is controlled with reference to the section 5.4.7 Power Supply Mode Control Protection Register (PMUKEY) and the 5.4.8 Power Supply Mode Control Register (PMUCTL).

Figure 23-5: Flowchart of entering deep sleep mode with partial power down



Note: When the interrupt request flag is “1” (an interrupt request signal is generated), the interrupt request signal is used to release the deep sleep mode. Therefore, if the WFI instruction is executed in this case, it is released as soon as the deep sleep mode is entered, and the partial power-down mode is not entered in this case. The WFI command is executed and returns to the operation mode after the deep sleep mode release time.

Table 23-3: Operation status in deep sleep mode with partial power down

| Item | | Deep sleep mode with partial power down |
|-----------------------------------------------------------------|---------------------|------------------------------------------------------------------------------------------------------------|
| System clock | F_{IH} | Operation stopped |
| | F_{IL} | The operating state is set via the OSMC register and the SUBCKSEL register, and the set state is retained. |
| CPU | | Operation stopped |
| Code flash memory | | Operation stopped |
| RAM | | Operation stopped |
| Port (latch) | | Status before deep sleep mode was set is retained |
| General-purpose timer unit TIMER4 | | Operation disabled |
| LSI_Timer | | Operable |
| Clock output/buzzer output | | Operation stopped |
| Watchdog timer | | Operable |
| DIVSQRT Unit | | Calculation stopped |
| TIMER0/1 | | Operation disabled |
| Capture/Compare/PWM (CCP0/1) | | Operation disabled |
| Enhanced PWM(EPWM) | | Operation disabled |
| UART | | Operation disabled |
| I ² C Serial Interface Controller (I ² C) | | Operation disabled |
| Serial Peripheral Interface Controller (SSP/SPI) | | Operation disabled |
| Analog-to-digital conversion (ADC) | | Operation disabled |
| Programmable Gain Amplifier (PGA0/1/2) | | Operation disabled |
| Analog comparator (ACMP0/1) | | Operation disabled |
| DAC | | Operation disabled |
| ADCLDO | | Operation disabled |
| Power-on reset function | | Operable |
| Voltage detection function | | Operable |
| External interrupt | | Operable |
| CRC | High-speed CRC | Operation stopped |
| | General-purpose CRC | Operation stopped |
| SFR guard function | | Operation stopped |

Note: Operation stopped: Operation is automatically stopped before switching to the deep sleep mode with partial power down.

Operation disabled: Operation is stopped before switching to the deep sleep mode with partial power down.

After shifting to a deep sleep mode with partial power down, the power supply to the module is stopped, and the module needs to be re-initialized after being released from the mode.

F_{IH} : High-speed on-chip oscillator clock F_{IL} : Low-speed on-chip oscillator clock

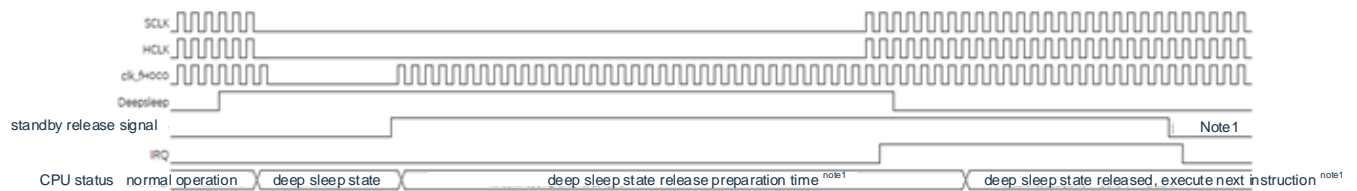
23.4.2 Release deep sleep mode with partial power down

Release the deep sleep mode with partial power down by the following 2 methods.

(a) Release deep sleep mode with partial power-down via interrupt requests

If INTP0-3, LSITIMER timer interrupt, LVI interrupt and WDT interrupt are requested, it is possible to release the deep sleep mode with partial power down. After the oscillation stabilization time, if it is allowed to accept interrupt, it will process the vector interrupt. If the interrupt is disabled, the next address is executed.

Figure 23-6: Release deep sleep mode by interrupting requests



Note 1: Standby release signals: INTP0-3, LSITIMER timer interrupt, LVI interrupt and WDT interrupt request signal.

Note 2: When the deep sleep state of partial power-down is ready to be released:

It is necessary to re-initialize the peripheral functions in order to ensure that the program continues to run normally.

(b) Release by generating a reset signal

The deep sleep mode with partial power-down is released by generating a reset signal. Then, as with a normal reset, the program is executed after switching to the reset vector address.

Figure 23-7: Release deep sleep mode with partial power-down by resetting



Note: For reset processing, refer to “Chapter 24 Reset Function”. Refer to “Chapter 25 Power-On Reset Circuit” for reset processing of the power-on reset (POR) circuit and voltage detection (LVD) circuit.

Chapter 24 Reset Function

The following six operations are available to generate a reset signal.

- (1) External reset input via RESETB pin.
- (2) Internal by watchdog timer program loop detection.
- (3) Internal reset by comparison of the supply voltage and detection voltage of power-on reset (POR) circuit.
- (4) Internal reset by comparison of supply voltage of the voltage detection circuit (LVD) and detection voltage.
- (5) Internal reset by setting the system reset request register bit (AIRC.R.SYSRESETREQ) to 1.
- (6) Internal reset by illegal memory access.

Internal reset is the same as external reset, and after generating a reset signal, the program is executed starting from the user-defined program start address.

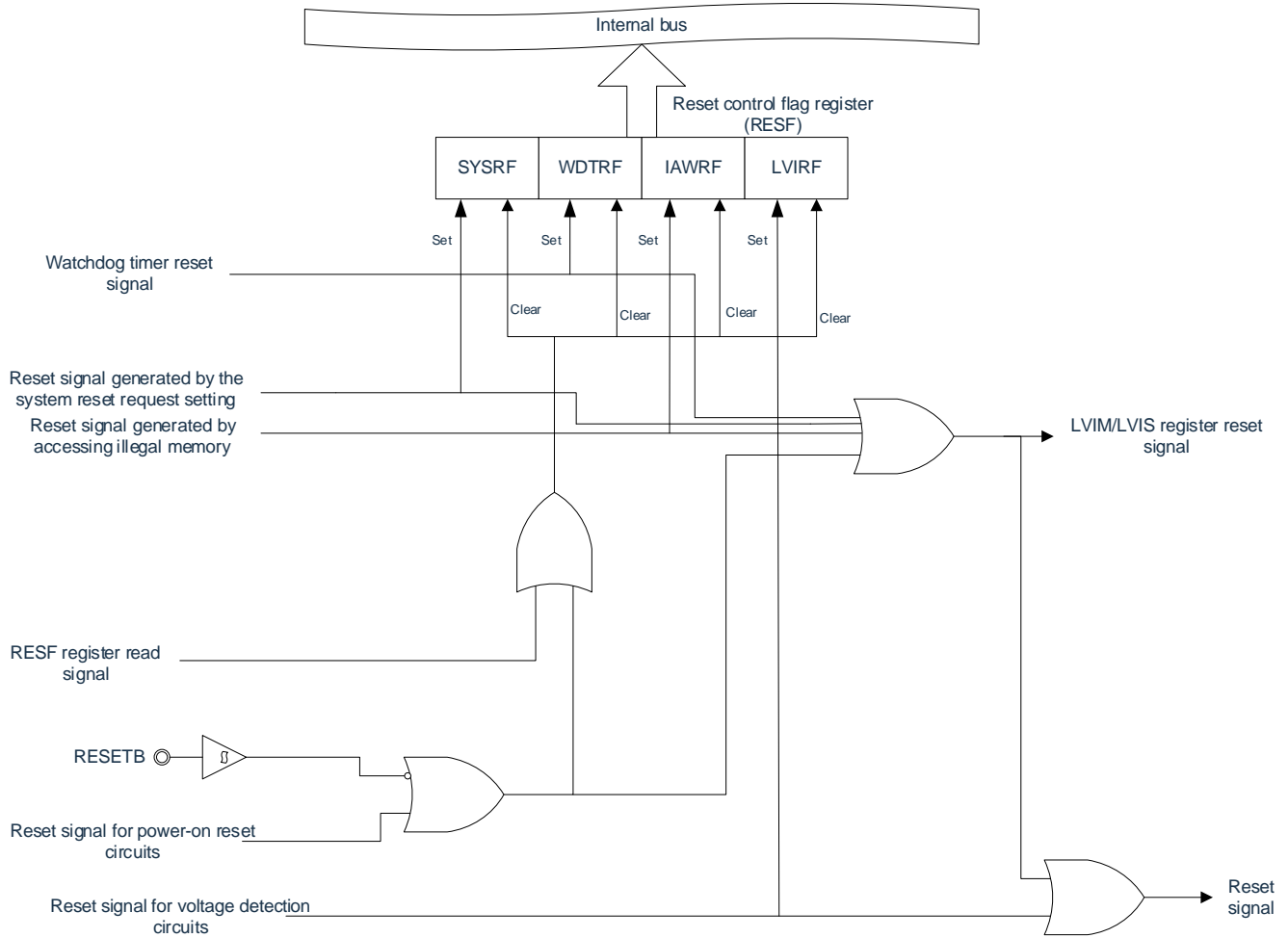
When a low level is supplied to the RESETB pin, or a program runaway is detected by the watchdog timer, or a voltage is detected in the POR and LVD circuits, or the system reset request bit is set, or an illegal memory access occurs, a reset is generated and the hardware changes to the state shown in Table 24-1.

Note 1: When performing an external reset, the RESETB pin must be held low for at least 10us. If an external reset is performed while the supply voltage is rising, the power must be turned on after supplying a low level to the RESETB pin, and must be held low for at least 10us over the operating voltage range shown in the AC Characteristics of the User's Manual, and then be supplied with a high level

Note 2: If a reset occurs, each SFR is initialized so that the pins change to the following states:

- During an external or power-on reset, except for P02 with internal pull-up and P06, P07 with internal pull-down, all other I/Os are in a high-impedance state.

Figure 24-1: Block diagram of reset function



Note 1: An internal reset of the LVD circuit does not reset the LVD circuit.

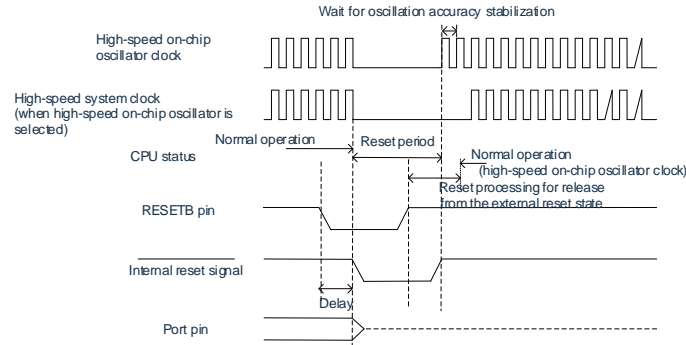
Note 2: LVIM: Voltage detection register

Note 3: LVIS: Voltage detection level register

1. Reset timing

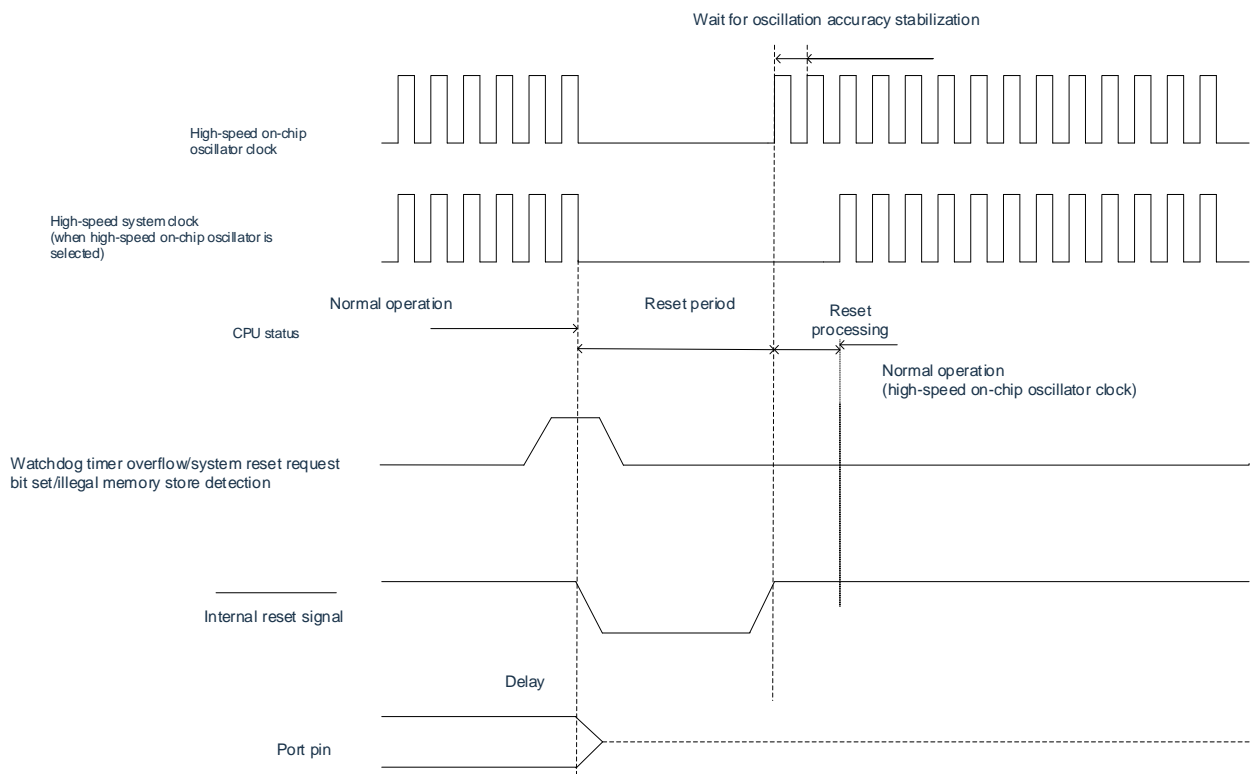
When the RESETB pin is input low, a reset is generated. The reset state is then released if the RESETB pin is input high and the program begins with a high-speed on-chip oscillator clock after the reset process is completed.

Figure 24-2: Reset timing of RESETB input



Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of a system request, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 24-3: Timing of reset due to watchdog timer overflow, set of system reset request bit, or detection of illegal memory access



Note 1: The port pins P02, P06, and P07 have the following states:

- During an external reset, P02 is determined by the external input signal, while P06 and P07 are at a low level.

- During a power-on reset, P02 is at a high level, while P06 and P07 are at a low level.

Note 2: The watchdog timer is also reset when an internal reset occurs.

For resets generated by the POR circuit and LVD circuit voltage detection, if $VDD \geq V_{POR}$ or $VDD \geq V_{LVD}$ after the reset, the reset state is released, and the program starts executing using the high-speed on-chip oscillator clock after the reset processing. For more details, please refer to “Chapter 24 Power-On Reset Circuit” and “Chapter 26 Voltage Detection Circuit”.

Note 3: V_{POR} : POR supply voltage rising detection voltage

V_{LVD} : LVD detection voltage

Table 24-1: Operation status during resetting

| Item | | Reset period | | | | | |
|--------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| | | External reset RESINB | Watchdog reset | Power-on reset | LVD reset | Write reset register reset | Illegal memory access reset |
| System clock | F _{IH} | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| | F _{IL} | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| CPU | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Code flash memory | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| RAM | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Port (latch) | | P02 is high after power on, P06, P07 is low, and other pins are in high resistance state before and after reset. | P02 is high after power on, P06, P07 is low, and other pins are in high resistance state before and after reset. | P02 is low, other pins are in high resistance state. | Without configuration, P02 is high after power on, P06, P07 is low, and all other pins are in high resistance state before and after reset. | P02 is high after power on, P06, P07 is low, and other pins are in high resistance state before and after reset. | P02 is high after power on, P06, P07 is low, and other pins are in high resistance state before and after reset. |
| General-purpose timer unit TIMER4 | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| LSI_Timer | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Clock output/buzzer output | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Watchdog timer | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| DIVSQRT Unit | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| TIMER0/1 | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Capture/Compare/P WM (CCP0/1) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Enhanced PWM(EPWM) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| UART | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| I ² C Serial Interface Controller (I ² C) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Serial Peripheral Interface Controller (SSP/SPI) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Analog-to-digital conversion (ADC) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Programmable Gain Amplifier (PGA0/1/2) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Analog comparator (ACMP0/1) | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |

| | | | | | | | |
|-------------------------------------------------------------------|-----------------------|--------------------|-------------------|-----------------------------------|-------------------|--------------------|--------------------|
| DAC | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| ADCLDO | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Power-on reset function | | Operable | Operable | Operable | Operable | Operable | Operable |
| Voltage detection function | | Operation disabled | | | Operable | Operation disabled | Operation disabled |
| External interrupt | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| CRC | High-speed CRC | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| | General - purpose CRC | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| SFR guard function | | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped | Operation stopped |
| Power-on state after reset completion (option byte configuration) | | Re-configure | Re-configure | Perform option byte configuration | Re-configure | Re-configure | Re-configure |

Note: P02 is high after power-on (in external input state during external reset), P06,P07 are low, and all other pins are in high resistance state before and after reset.

F_{IH}: High-speed internal clock

F_{IL}: Low-speed internal clock

24.1 Registers for confirming the reset source

24.1.1 Register mapping

(Reset control base address = 0x4002_0440) RO: Read only, WO: Write Only, R/W: Read/Write.

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-----------------------------|-------------|
| RESF | 0x000 | RO | Reset control flag register | - |

24.1.2 Reset control flag register (RESF)

The CMS32M65xx microcontroller has multiple internal reset generation sources. The Reset Control Flag register (RESF) holds the reset source where the reset request occurs. The RESF register can be read by an 8-bit memory manipulation instruction.

The SYSRF, WDTRF, IAWRF, LVIRF flags are cleared by inputting RESETB, resetting the power-on reset (POR) circuit, and reading the RESF register. To determine the reset source, the value of the RESF register must be saved to any RAM and then determined by its RAM value.

| Bit | Symbol | Description | Reset value ^{Note1} |
|-----|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| 7 | SYSRF | Internal reset request resulting from the system reset request bit being set 0: No internal reset request is generated or the RESF register is cleared. 1: An internal reset request is generated. | - |
| 6:5 | - | Reserved | - |
| 4 | WDTRF | Internal reset request generated by the watchdog timer (WDT) 0: No internal reset request is generated or the RESF register is cleared. 1: An internal reset request is generated. | - |
| 3:2 | - | Reserved | - |
| 1 | IAWRF | Access to internal reset requests generated by illegal memory 0: No internal reset request is generated or the RESF register is cleared. 1: An internal reset request is generated. | - |
| 0 | LVIRF | Internal reset request generated by the voltage | - |

| | | | |
|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| | | detection circuit (LVD) 0: No internal reset request is generated or the RESF register is cleared. 1: An internal reset request is generated. | |
|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

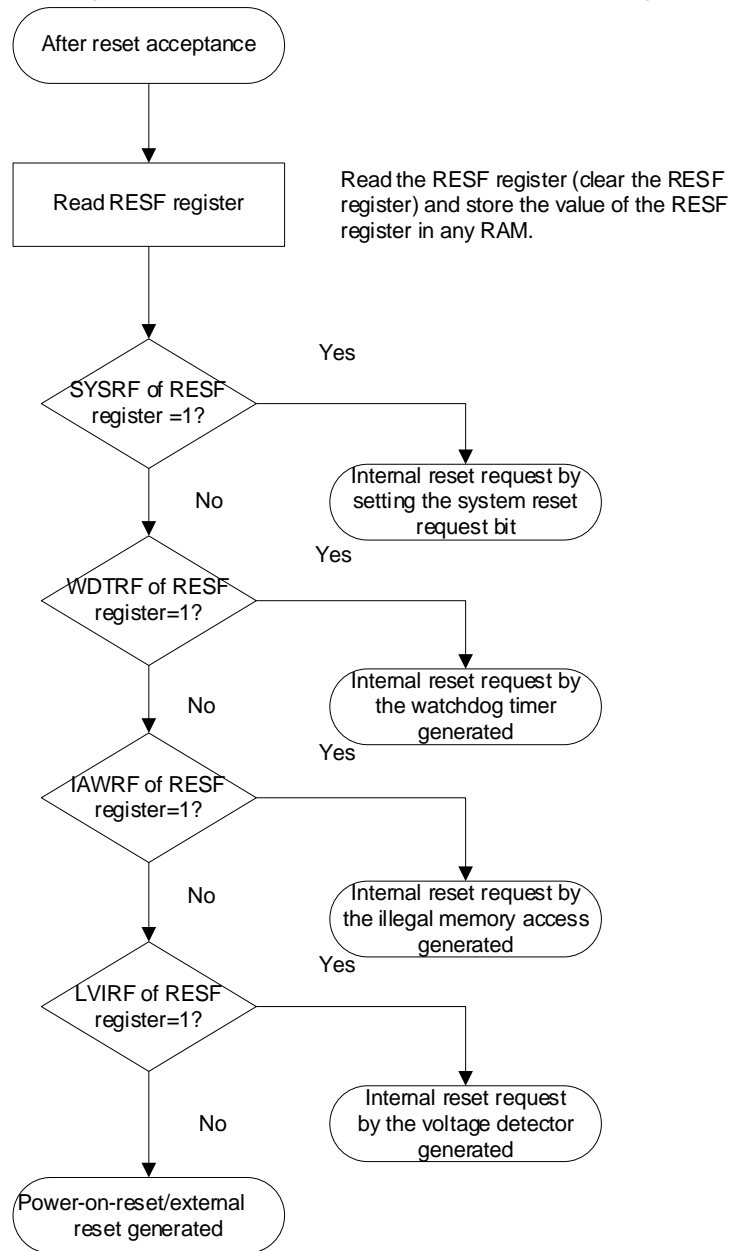
Note: The value after reset varies depending on the reset source. See Table 24-2.

Table 24-2: RESF register status when a reset request occurs

| Reset source Flag | RESETB input | Reset by POR | Reset generated by system reset request bit | Reset generated by WDT | Reset generated by accessing illegal | Reset generated by LVD |
|----------------------|-----------------|-----------------|------------------------------------------------------|------------------------------|-----------------------------------------------|------------------------------|
| SYSRF | Cleared to "0" | Cleared to "0" | Set to "1" | Held | Held | Held |
| WDTRF | | | Held | Set to "1" | Set to "1" | |
| IAWRF | | | | | Held | |
| LVIRF | | | | | Held | Set to "1" |

Figure 24-4 shows the procedure for checking a reset source.

Figure 24-4: Example of procedure for checking reset source



Note: The flow described above is an example of the procedure for checking.

Chapter 25 Power-On-Reset Circuit

25.1 Function of power-on-reset circuit

The power-on-reset circuit (POR) has the following functions.

- ① Generates internal reset signal at power on.

The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal.

- ② Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power is supplied, this should be placed in the deep sleep mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in AC Characteristics. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Note 1: When the power-on reset circuit generates an internal reset signal, the reset control flag register (RESF) is cleared to “00H”.

Note 2: The CMS32M65xx microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), system reset request bit setting, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), system reset request bit setting, or illegal-memory access. For details of RESF register, refer to “Chapter 24 Reset Function”.

Note 3: V_{POR} : POR power supply rise detection voltage

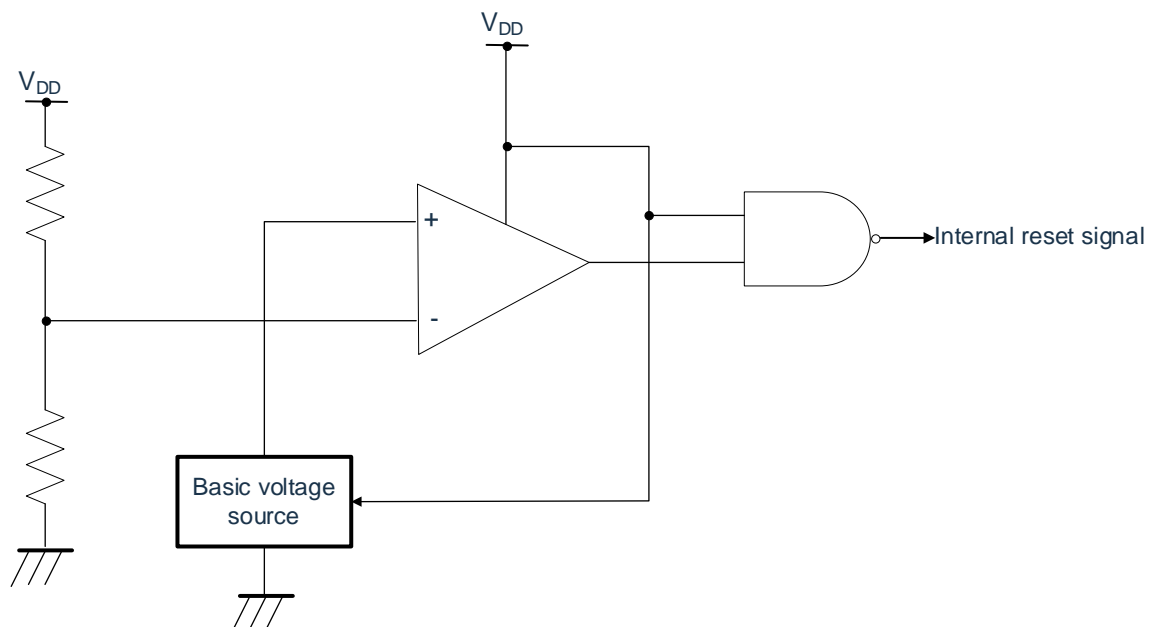
V_{PDR} : POR power supply fall detection voltage

For details, refer to the POR circuit characteristics in the data sheet.

25.2 Structure of power-on reset circuit

The block diagram of the power-on reset circuit is shown in Figure 25-1.

Figure 25-1: Block diagram of power-on reset circuit

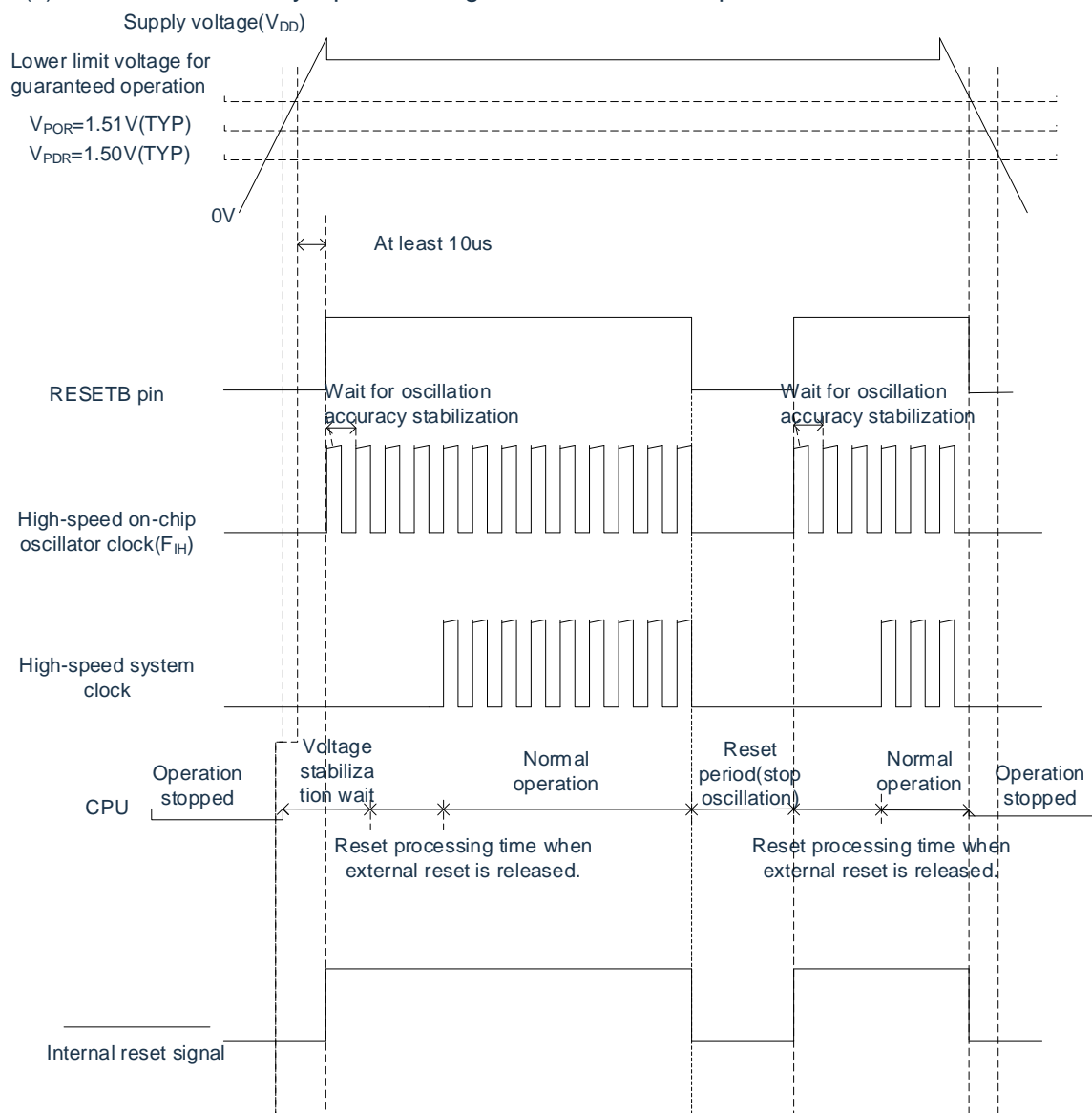


25.3 Operation of power-on reset circuit

The timing of the internal reset signal generation for the power-on reset circuit and the voltage detection circuit is shown below.

Figure 25-2: Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (1/3)

(1) When the externally input reset signal on the RESETB pin is used



Note 1: The internal reset processing time includes the oscillation accuracy stabilization wait time of the high-speed on-chip oscillator clock.

Note 2: When the power supply voltage rises, the power supply voltage must be maintained by external reset before it reaches the working voltage range shown in the AC characteristics of the data sheet; When the supply voltage drops, it must be reset through deep sleep mode transfer, voltage detection circuitry, or external reset before the supply voltage falls below the operating voltage range. When restarting operation, you must confirm that the supply voltage has returned to the

operating voltage range.

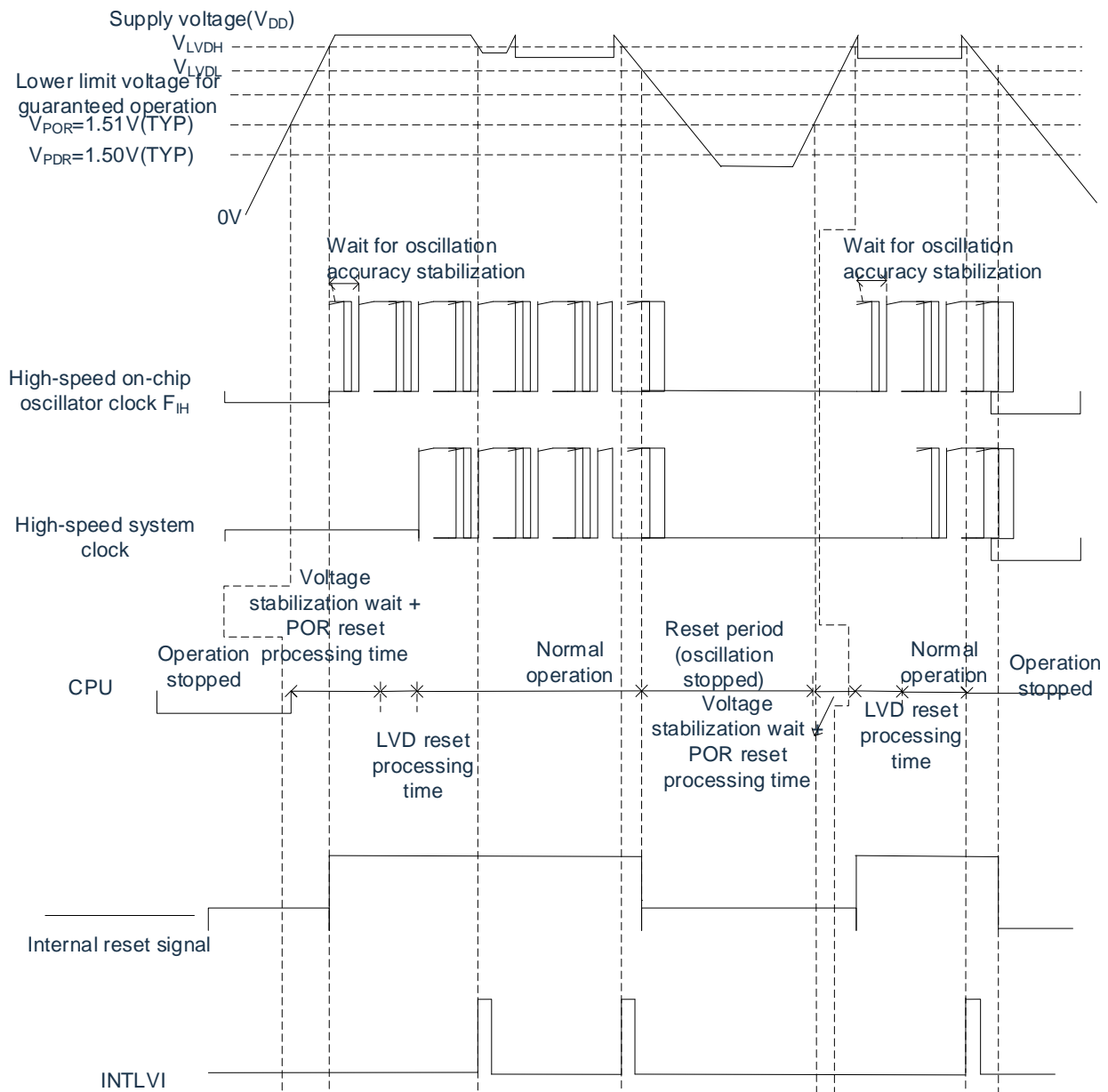
Note 3: VPOR: POR power supply rise detection voltage

VPDR: POR power supply fall detection voltage

Note 4: When LVD is OFF, the external reset of RESETB pin must be used. For details, please refer to “Chapter 26 Voltage Detection Circuit”.

Figure 25-2: Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (2/3)

(2) LVD interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0=1, 0)



Note 1: The internal reset processing time includes the oscillation accuracy stabilization wait time of the high-speed on-chip oscillator clock.

Note 2: After generating the interrupt request signal (INTLVI), the LVIV bit and the LVIMD bit of the voltage detection level register (LVIS) are automatically set to “1”. Therefore, considering the

possibility that the power supply voltage may return to the high voltage detection voltage (V_{LVDH}) or higher without falling below the low voltage detection voltage (V_{LVDL}), follow the steps in “Figure 26-5 Setting Procedure for Confirmation/Reset of Operating Voltage” and “Figure 26-6 Setting Procedure for Interrupt and Reset” after generating an INTLVI.

Note 3: The time until normal operation begins includes the “Voltage Stabilization Wait + POR Reset Processing Time” after V_{POR} (1.51V (TYP.)) is reached as well as the “LVD Reset Processing Time” after the LVD detection level (V_{LVDH}) is reached.

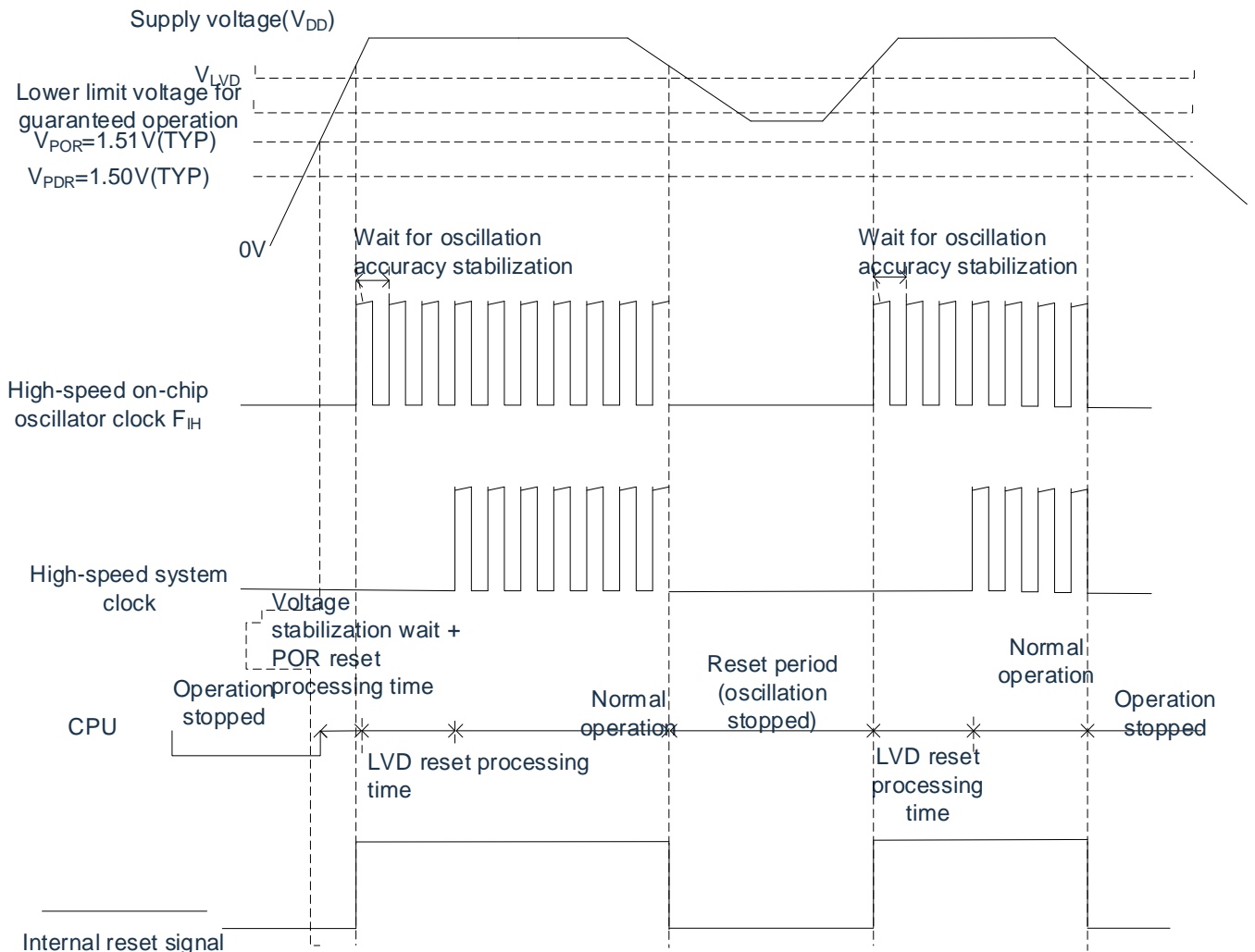
Note 4: V_{LVDH} , V_{LVDL} : LVD detection voltage

V_{POR} : POR power supply rise detection voltage

V_{PDR} : POR power supply fall detection voltage

Figure 25-2: Timing of internal reset signal generation for power-on reset circuit and voltage detection circuit (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1 = 1, LVIMDS0 = 1)



Note 1: The internal reset processing time includes the oscillation accuracy stabilization wait time of the high-speed on-chip oscillator clock.

Note 2: The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V(TYP.)) is reached.

Note 3: When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detection circuit (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.

Note 4: V_{LVDH} , V_{LVLDL} : LVD detection voltage

V_{POR} : POR supply voltage rise detection voltage

V_{PDR} : POR supply voltage fall detection voltage

Note 5: When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is powered on is the same as the time specified in Figure 25-2 (3/3).

Chapter 26 Voltage Detection Circuit

26.1 Function of voltage detection circuit

The voltage detection circuit sets the operating mode and detection voltage (V_{LVDH} , V_{LVDL} , V_{LVD}) by option byte (000C1H). The voltage detection circuit (LVD) has the following functions.

- 1) The internal reset or internal interrupt signal is generated by comparing the supply voltage (V_{DD}) with the detection voltage (V_{LVDH} , V_{LVDL} , V_{LVD}).
- 2) The detection voltage of the supply voltage (V_{LVDH} , V_{LVDL}) can be selected from 12 detection levels by means of option bytes (see “Chapter 29 Option Byte”).
- 3) It can also operate in deep sleep mode.
- 4) When the supply voltage rises, the reset state must be maintained by the voltage detection circuit or external reset before the supply voltage reaches the operating voltage range shown in the AC characteristics of the datasheet; when the supply voltage falls, the reset state must be set by the deep sleep mode transfer, voltage detection circuit or external reset before the supply voltage falls below the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H).

(a) Interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0 of option byte)

Two detection voltages (V_{LVDH} , V_{LVDL}) are selected by the option byte 000C1H. The high voltage detection level (V_{LVDH}) is used to release the reset or generate an interrupt, and the low voltage detection level (V_{LVDL}) is used to generate a reset.

(b) Reset mode (LVIMDS1, LVIMDS0=1, 1 for option byte)

A detection voltage (V_{LVD}) selected by option byte 000C1H is used to generate or release the reset.

(c) Interrupt mode (option byte of LVIMDS1, LVIMDS0=0, 1)

A detection voltage (V_{LVD}) selected by option byte 000C1H is used to generate an interrupt or to release the reset. In each mode, the following interrupt signals and internal reset signals are generated.

| Mode | Interrupt & reset mode | Reset mode | Interrupt mode |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuration | (LVIMDS1, LVIMDS0=1, 0) | (LVIMDS1, LVIMDS0=1, 1) | (LVIMDS1, LVIMDS0=0, 1) |
| Operation process | When the operating voltage drops, an interrupt request signal is generated when $V_{DD} < V_{LVDH}$ is detected; when $V_{DD} < V_{LVDL}$ is detected, an internal reset is generated. When $V_{DD} \geq V_{LVDH}$ is detected, an internal reset is released. | When $V_{DD} \geq V_{LVD}$ is detected, an internal reset is released; when $V_{DD} < V_{LVD}$ is detected, an internal reset is generated. | After a reset occurs, an internal reset state of LVD continues until $V_{DD} \geq V_{LVD}$. When $V_{DD} \geq V_{LVD}$ is detected, an internal reset of LVD is released. After the internal reset of LVD is released, if $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected, then an interrupt request signal (INTLVI) is generated. |

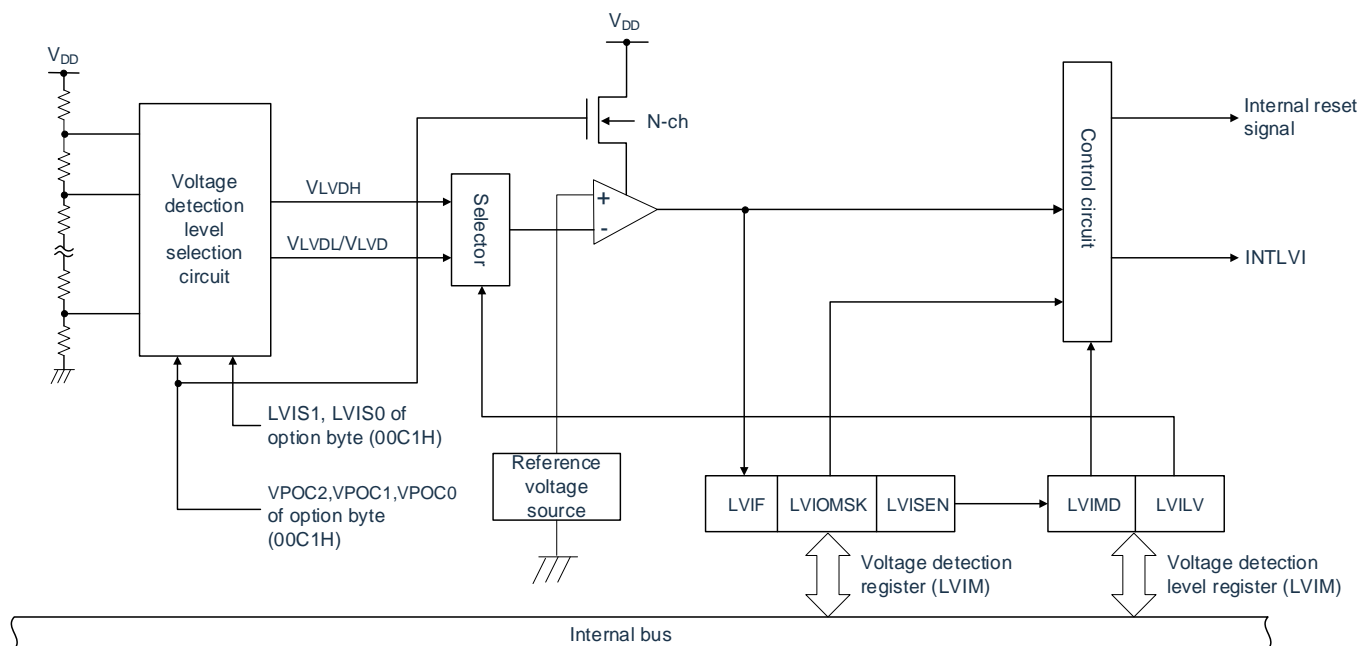
When the voltage detection circuit is in operation, it is possible to check whether the power supply voltage is greater than or less than the detection voltage by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

If a reset occurs, bit 0 (LVIRF) of the reset control flag register (RESF) is set to “1”. For details of the RESF register, please refer to “Chapter 24 Reset Function”.

26.2 Structure of voltage detection circuit

The block diagram of the voltage detection circuit is shown in Figure 26-1.

Figure 26-1: Block diagram of voltage detection circuit



26.3 Register mapping

(Voltage detection base address = 0x4002_0441)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|-------------------------------|-------------|
| LVIM | 0x000 | R/W | Voltage detection register | 0x0 |
| LVIS | 0x001 | R/W | Voltage detect level register | 0x0 |

26.4 Registers for controlling voltage detection circuit

The voltage detection circuit is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

26.4.1 Voltage detection register (LVIM)

This register is set to enable or disable overwriting of the voltage detection level register (LVIS), and to confirm the masking status of the LVD output. The LVIM register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes “00H”.

| Bit | Symbol | Description | Reset value |
|-----|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | LVISEN ^{Note1} | Enable/disable setting of voltage detection level register (LVIS) 0: Disable 1: Enable | 0 |
| 6:2 | - | Reserved | - |
| 1 | LVIOMSK | Mask status flag for LVD output 0: LVD output masking is invalid. 1: LVD output masking is valid. ^{Note2} | 0 |
| 0 | LVIF | Voltage detection flag 0: Supply voltage (V_{DD}) \geq detection voltage (V_{LVD}) or LVD is OFF. 1: Supply voltage (V_{DD}) $<$ detection voltage (V_{LVD}) | 0 |

Note 1. It can only be set when the interrupt & reset mode is selected (lvIMDS1 bits and LVIMDS0 bits of the option bytes are “1” and “0” respectively), the initial value cannot be changed in other modes.

2. Only when the interrupt & reset mode is selected (the LVIMDS1 bit and LVIMDS0 bits of the option byte are “1” and “0” respectively). The LVIOMSK bit automatically changes to “1” during the following periods, masking the reset or interrupt generated by LVD.

- When LVISEN=1
- Waiting time from the occurrence of LVD interrupt to the stabilization of LVD detection voltage
- Waiting time from changing the value of the LVILV bit (bit0 of the LVIS register) until the LVD detection voltage stabilizes.

26.4.2 Voltage detection level register (LVIS)

This is a register that sets the voltage detection level.

The LVIS register is set by an 8-bit memory manipulation instruction. After generating a reset signal, the value of this register changes to “00H/01H/81H” ^{Note1}.

| Bit | Symbol | Description | Reset value |
|-----|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | LVIMD ^{Note1} | Operation mode of voltage detection 0: Interrupt mode 1: Reset mode | 0 |
| 6:1 | - | Reserved | - |
| 0 | LVILV ^{Note2} | LVD detection level 0: High voltage detection level (V_{LVDH}) 1: Low voltage detection level (V_{LVDL} or V_{LVD}) | 0 |

Note 1. The reset value varies depending on the setting of the reset source and option bytes. When an LVD reset occurs, this register is not cleared to “00H”.

When a reset other than LVD occurs, the values of this register are as follows:

- LVIMDS1, LVIMDS0 of Option bytes =1, 0: 00H
- LVIMDS1, LVIMDS0 of Option bytes =1, 1: 81H
- LVIMDS1, LVIMDS0 of Option bytes =0, 1: 01H

2. Write “0” only if interrupt & reset mode is selected (LVIMDS1 bit and LVIMDS0 bits for option bytes are “1” and “0” respectively). In other cases, it cannot be set. In interrupt & reset mode, value substitution is performed automatically by generating a reset or interrupt.
3. To rewrite the LVIS registers, it must be done in accordance with the steps in Figure 26-5 and Figure 26-6.
4. Option byte 000C1H selects the mode of operation of the LVD and the detection voltage (V_{LVDH} , V_{LVDL} , V_{LVD}) for each mode. For details of the user option byte (000C1H/010C1H), refer to “Chapter 29 Option Byte”.

26.5 Operation of voltage detection circuit

26.5.1 When used as reset mode

The operation mode (reset mode (LVIMDS1, LVIMDS0=1, 1)) and the detection voltage (V_{LVD}) are set via the option byte 000C1H. If the reset mode is set, operation starts with the following initial settings.

- Set bit 7 (LVISEN) of the voltage detection register (LVIM) to “0” (disable rewriting the voltage detection level register (LVIS))
- Set the initial value of the voltage detection level register (LVIS) to “81H”. Set bit7(LVIMD) to “1” (reset mode). Set bit0 (LVILV) to “1” (voltage detection level: V_{LVD}).

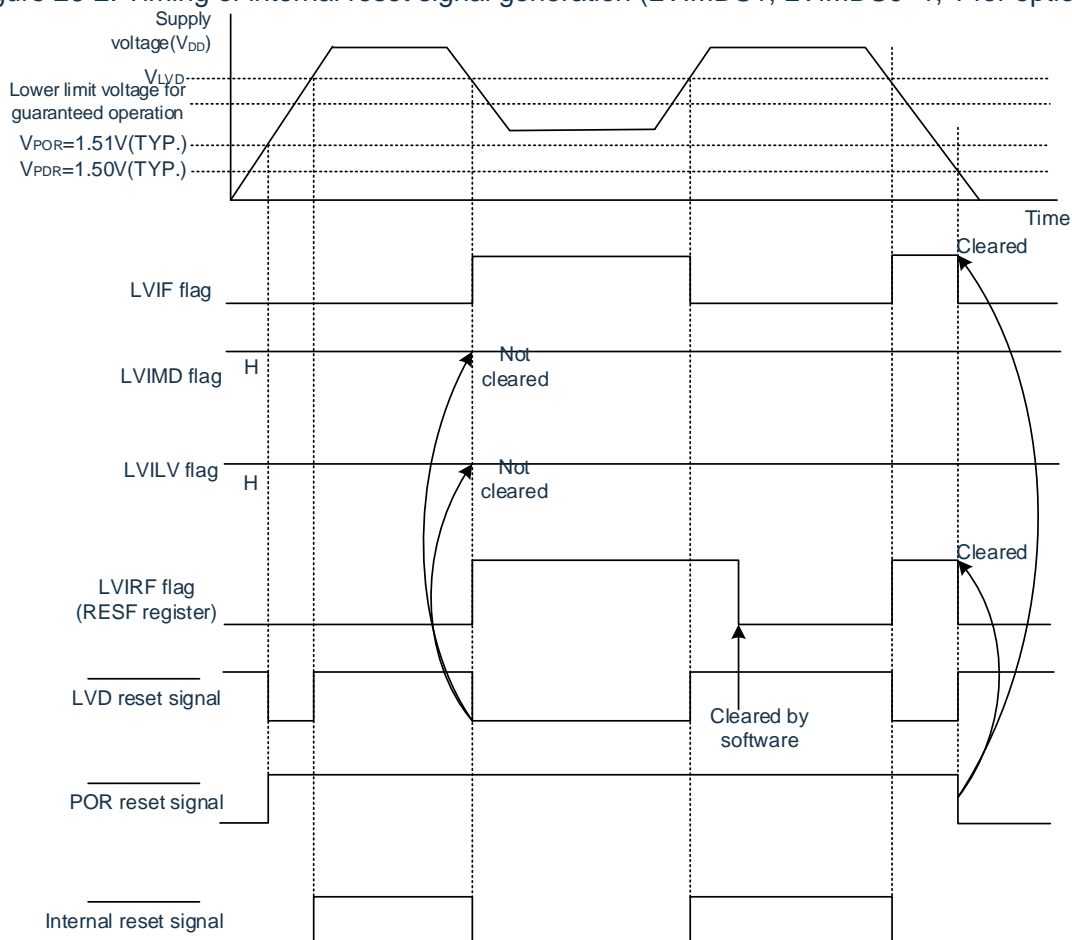
● Operation of LVD reset mode

When the power is turned on, the reset mode (LVIMDS1, LVIMDS0=1, 1 of the option byte) keeps the internal reset state of LVD until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). If the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}), the internal reset is released.

When the operating voltage falls, an internal reset of LVD is generated if the supply voltage (V_{DD}) is below the voltage detection level (V_{LVD})

The timing of the internal reset signal generation for LVD reset mode is shown in Figure 26-2.

Figure 26-2: Timing of internal reset signal generation (LVIMDS1, LVIMDS0=1, 1 for option byte)



Note : VPOR: POR power supply rise detection voltage

VPDR: POR power supply fall detection voltage

26.6 When used as interrupt mode

The operation mode (interrupt mode (LVIMDS1, LVIMDS0=0, 1)) and the detection voltage (V_{LVD}) are set via the option byte 000C1H. If the interrupt mode is set, operation starts with the following initial settings.

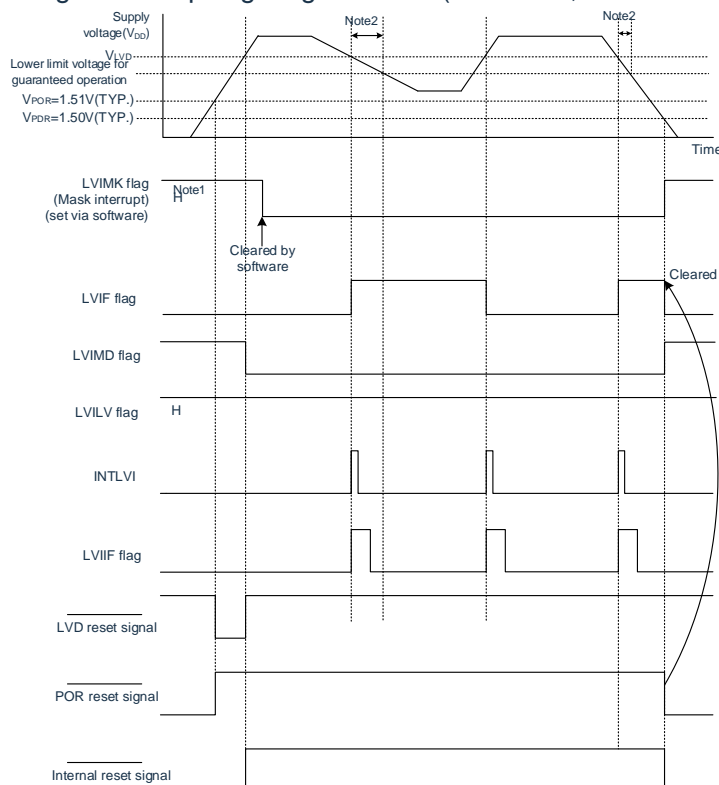
- Set bit 7 (LVISEN) of the voltage detection register (LVIM) to “0” (disables rewriting the voltage detection level register (LVIS)).
- Set the initial value of the voltage detection level register (LVIS) to “01H”. Set bit7 (LVIMD) to “0” (interrupt mode). Set bit0(LVILV) to “1” (voltage detection level: VLVD).
- Operation of LVD interrupt mode

After generating a reset, the interrupt mode (LVIMDS1, LVIMDS0 of the option byte =0, 1) maintains the internal reset state of the LVD until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). If the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}), the internal reset of the LVD is released.

If the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after the internal reset of the LVD is released, an interrupt request signal (INTLVI) of the LVD is generated. When the operating voltage drops, it must be set to the reset state by deep sleep mode transfer or external reset before the operating voltage falls below the operating voltage range shown in the AC characteristics of the datasheet. When restarting operation, it must be verified that the supply voltage has returned to the operating voltage range.

The timing of the interrupt request signal generation for LVD interrupt mode is shown in Figure 26-3

Figure 26-3: Timing of interrupt signal generation (LVIMDS1, LVIMDS0 of option byte =0, 1)



Note 1. After generating a reset signal, the LVIMK flag changes to “1”.

2. When the operating voltage drops, it must be reset by deep sleep mode transfer or external reset before the operating voltage falls below the operating voltage range shown in the AC characteristics of the data sheet. When restarting operation, it must be verified that the supply

voltage returns to the operating voltage range.

3.VPOR: POR power supply rise detection voltage

VPDR: POR power supply fall detection voltage

26.7 When used as interrupt & reset mode

The operation mode (interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0)) and the detection voltage (V_{LVDH} , V_{LVDL}) are set via the option byte 000C1H.

If the interrupt & reset mode is set, the operation starts with the following initial settings.

- Set bit 7 (LVISEN) of the voltage detection register (LVIM) to “0” (disables rewriting the voltage detection level register (LVIS)).
- Set the initial value of the voltage detection level register (LVIS) to “00H”. Set bit7 (LVIMD) to “0” (interrupt mode). Set bit0(LVILV) to “0” (high voltage detection level: V_{LVDH}).

- Operation of LVD interrupt & reset mode

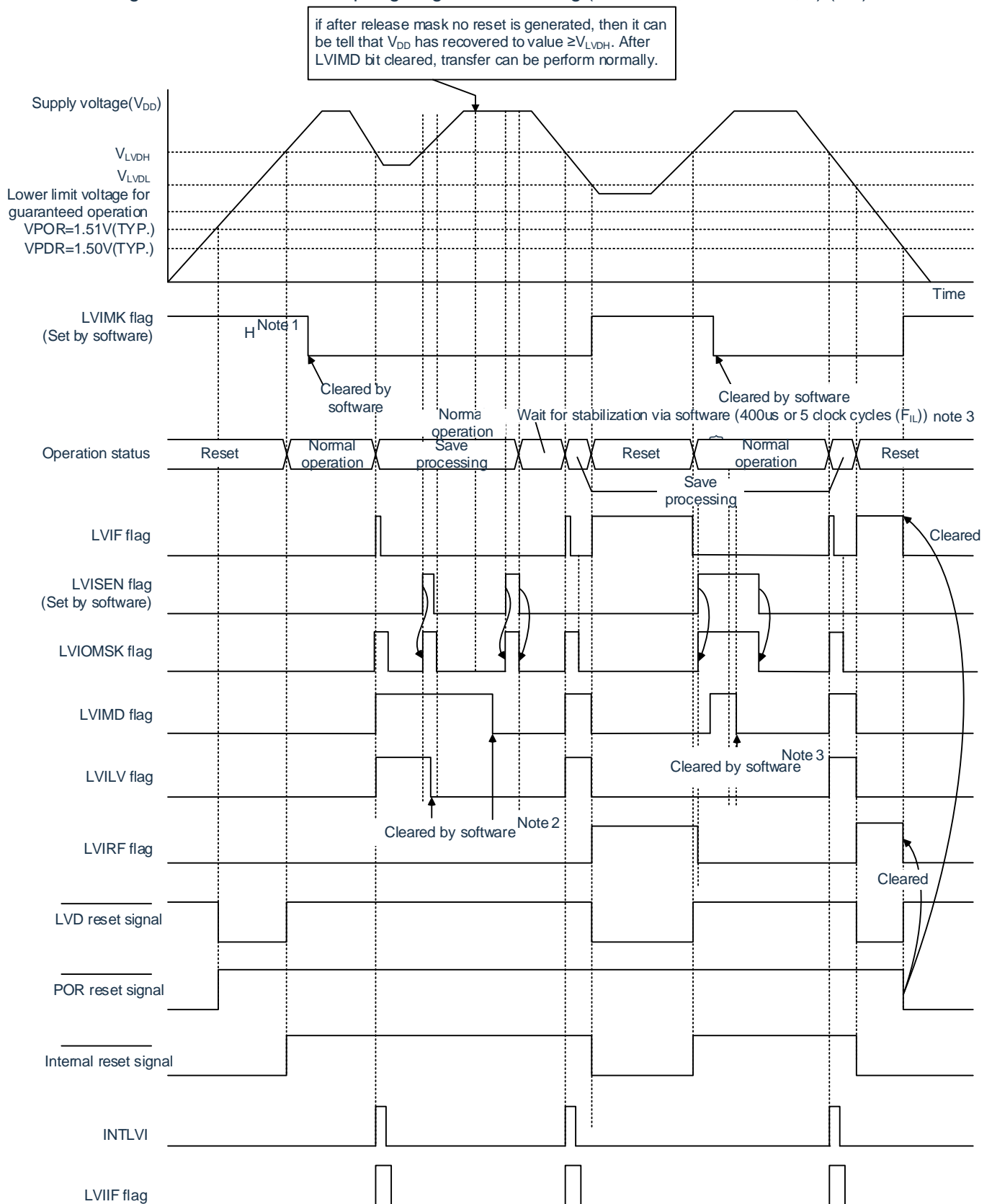
When power is turned on, the interrupt & reset mode (LVIMDS1, LVIMDS0=1, 0 of the option byte) maintains the internal reset state of the LVD until the power supply voltage (V_{DD}) exceeds the high voltage detection level (V_{LVDH}). If the supply voltage (V_{DD}) exceeds the high voltage detection level (V_{LVDH}), the internal reset is released.

When the operating voltage drops, if the supply voltage (V_{DD}) is below the high voltage detection level (V_{LVDH}), an interrupt request signal (INTLVI) is generated for the LVD and any stacking process can be performed. After that, if the supply voltage (V_{DD}) is below the low voltage detection level (V_{LVDL}), an internal reset of the LVD is generated. However, after INTLVI occurs, no interrupt request signal is generated even if the supply voltage (V_{DD}) returns to the high voltage detection voltage (V_{LVDH}) or higher without falling below the low voltage detection voltage (V_{LVDL}).

When using LVD interrupt & reset mode, you must follow “Figure 26-5: Setting procedure for confirmation /reset of operating voltage” and “Figure 26-6: Initial setting procedure for interrupt & reset mode”.

The timing of the internal reset signal and interrupt signal generation in LVD interrupt & reset mode is shown in Figure 26-4.

Figure 26-4: Reset & interrupt signal generation timing (LVIMDS1, LVIMDS0=1, 0) (1/2)



Note 1. After the reset signal is generated, the LVIMK flag becomes “1”.

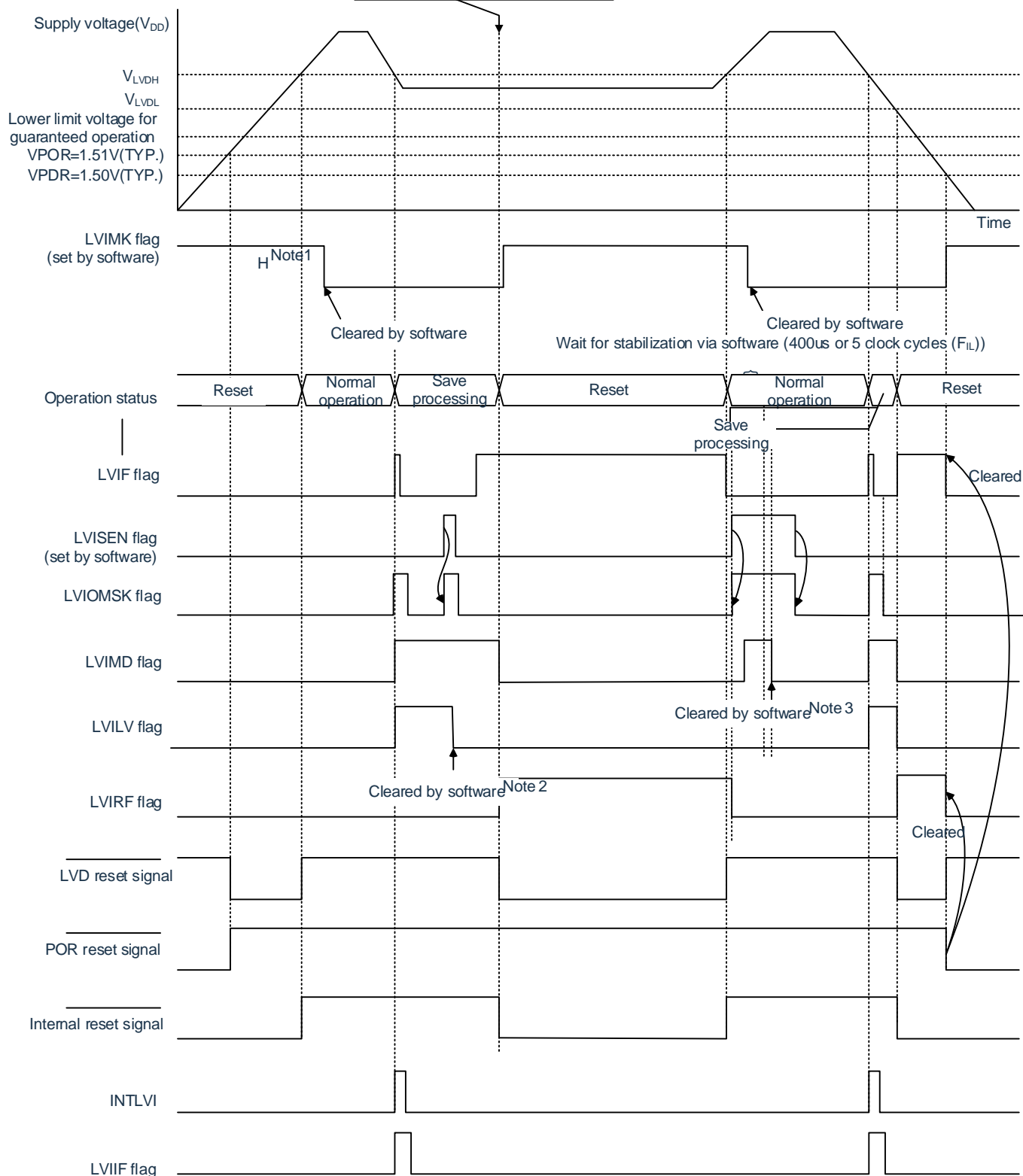
2. When using the interrupt & reset mode, you must follow “Figure 26-5: Setting procedure for confirmation /reset of operating voltage” after an interrupt occurs.

3. When using the interrupt&reset mode, you must follow the steps in “Figure 26-6: Initial setting procedure for interrupt & reset mode” after the reset is released.

4. V_{POR} : POR power supply rise detection voltage

V_{PDR} : POR power supply fall detection voltage

When a condition of V_{DD} is $V_{DD} < V_{LVIH}$ after releasing the mask, a reset is generated because of $LVIMD = 1$ (reset mode).



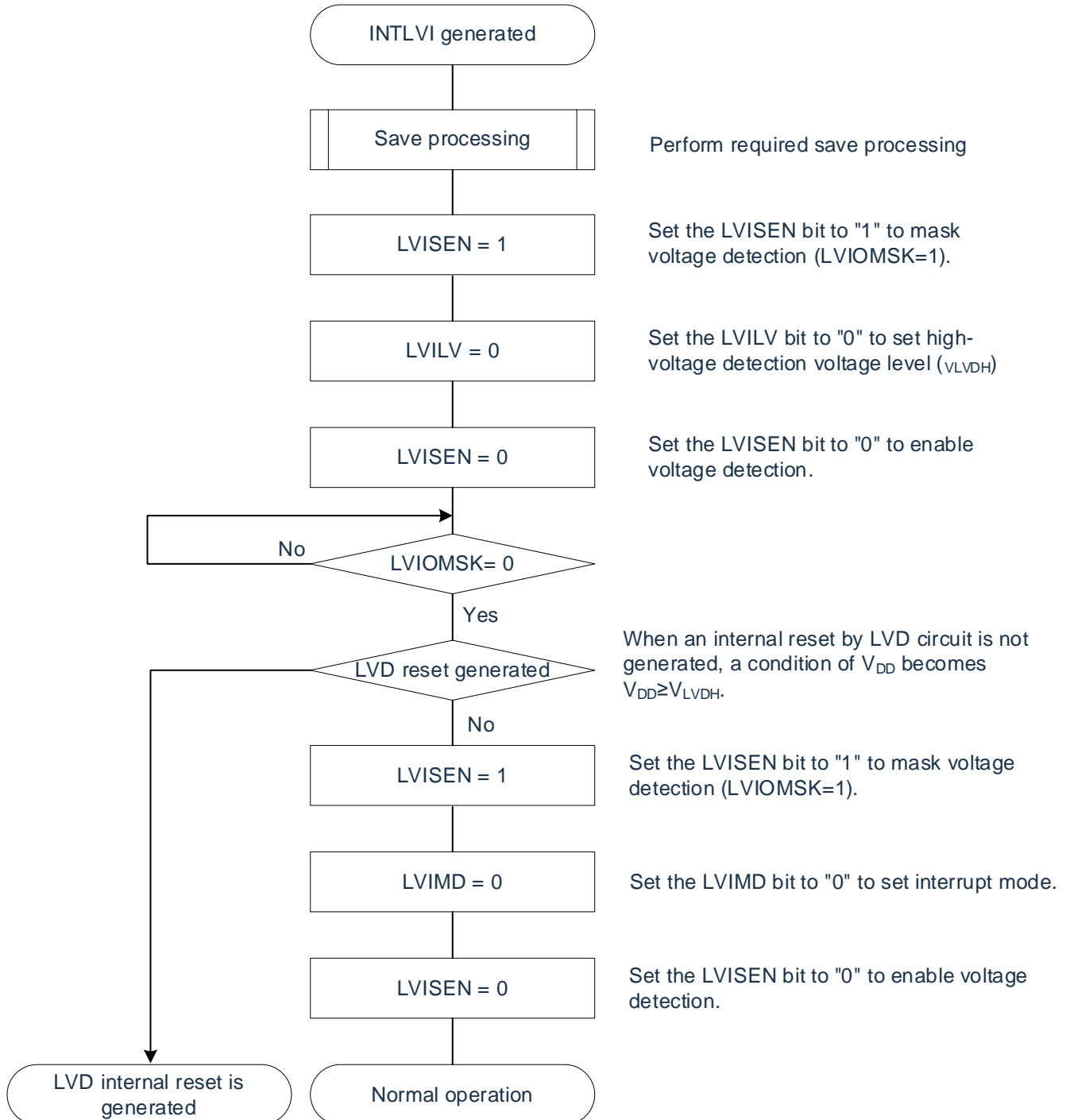
2. When using the interrupt & reset mode, you must follow “Figure 26-5: Setting procedure for confirmation /reset of operating voltage” after an interrupt occurs.

3. When using the interrupt&reset mode, you must follow the steps in “Figure 26-6: Initial setting procedure for interrupt & reset mode” after the reset is released.

4.VPOR: POR power supply rise detection voltage

VPDR: POR power supply fall detection voltage

Figure 26-5: Setting procedure for confirmation/reset of operating voltage

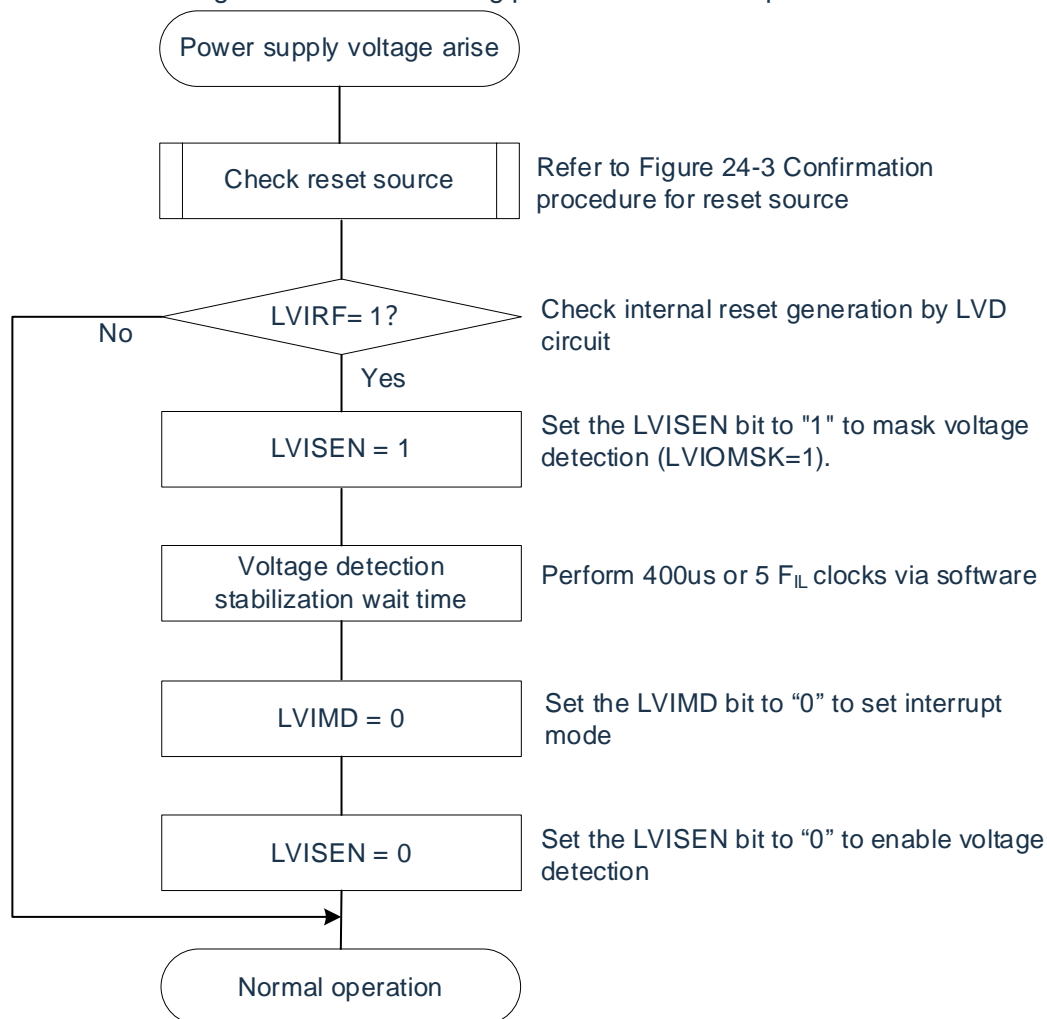


If the interrupt & reset mode is set (LVIMDS1, LVIMDS0=1, 0), it will take 400us or 5 F_{IL} clocks for the voltage detection to stabilize after the LVD reset (LVIRF=1) is released. The LVIMD bit must be cleared to "0" for initialization after waiting for the voltage detection to stabilize. The LVISEN bit must be set to "1" during the count of the voltage detection stabilization time and when rewriting the LVIMD bit to block the generation of

resets or interrupts generated by LVD.

The initial setting procedure for interrupt & reset mode is shown in Figure 26-6.

Figure 26-6: Initial setting procedure for interrupt & reset mode



Note: F_{IL} : Low-speed on-chip oscillator clock frequency

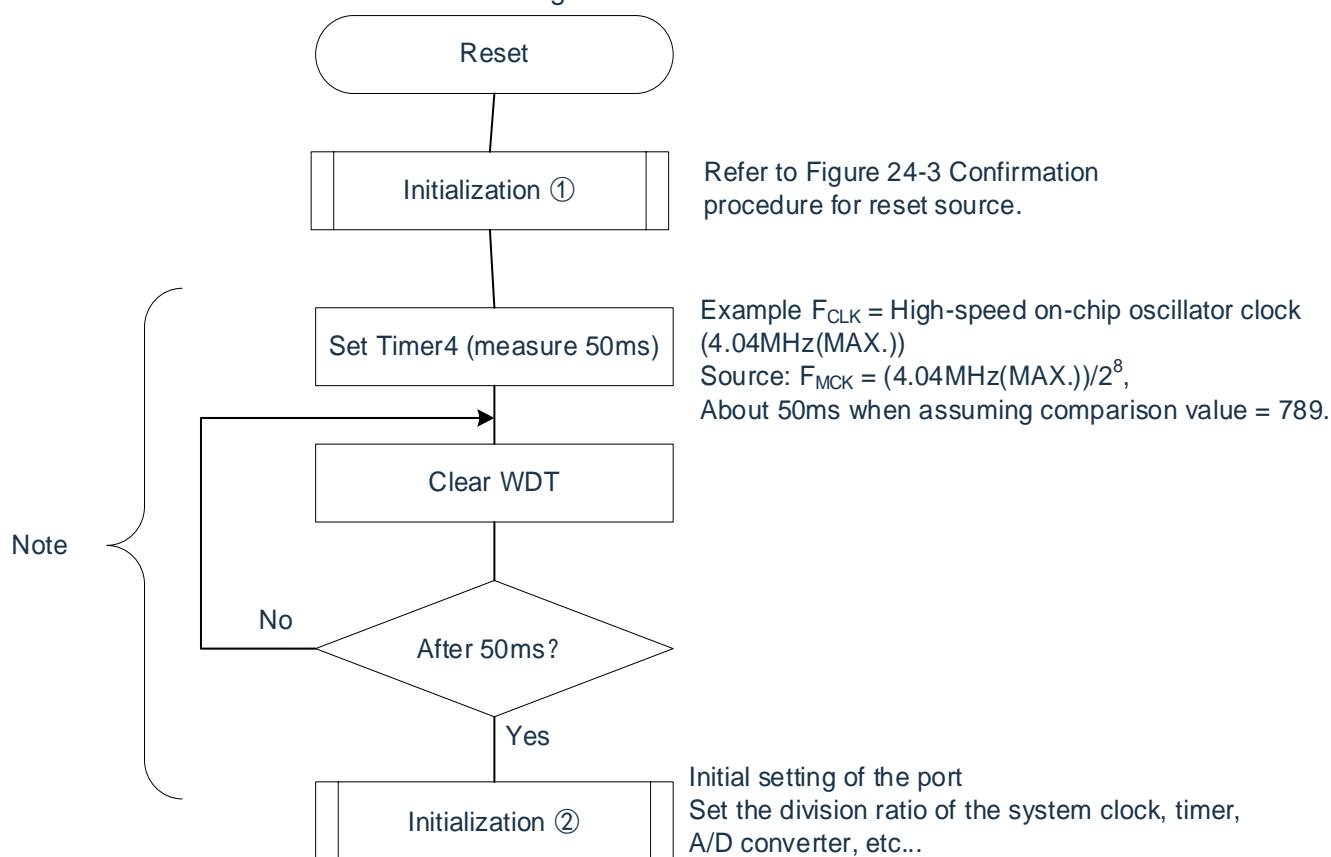
26.8 Cautions for voltage detection circuit

(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 26-7: Example of software processing when the supply voltage fluctuation near the LVD detection voltage does not exceed 50ms

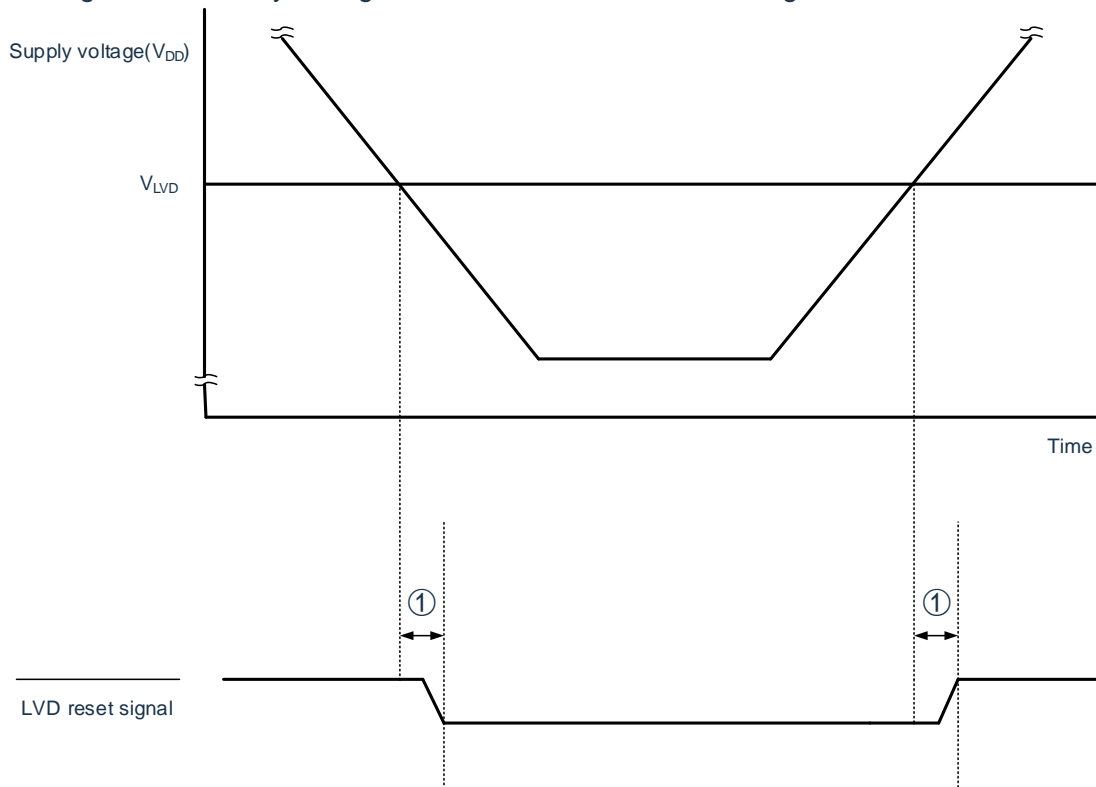


Note: If the reset occurs again during this period, it is not switched to initialization processing ②.

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

A delay occurs from the time the supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) is met to the time the LVD reset is generated. Similarly, a delay occurs from the time the LVD detection voltage (V_{LVD}) \leq the supply voltage (V_{DD}) to the time the LVD reset is released (see Figure 26-8).

Figure 26-8: Delay from generation of LVD reset source to generation or release of LVD reset



① Detection delay (300us(MAX.))

(3) When the power is turned on with LVD set to OFF

When LVD is set to OFF, an external reset must be performed using the RESETB pin.

When performing an external reset, the RESETB pin must be input low for at least 10us. If an external reset is performed while the supply voltage is rising, the power must be turned on after a low level is input to the RESETB pin, and must be held low for at least 10us within the operating voltage range shown in the AC characteristics of the datasheet, followed by a high level.

(4) When LVD is set to OFF in LVD interrupt mode and the operating voltage drops

If the operating voltage drops when LVD is set to OFF and LVD interrupt mode is set, it must be reset by deep sleep mode transfer or external reset before the operating voltage falls below the operating voltage range shown in the AC characteristics of the data sheet. When restarting operation, it is necessary to verify that the supply voltage is restored in the operating voltage range.

Chapter 27 Safety Functions

27.1 Overview of safety functions

The following safety functions are provided in the CMS32M65xx to comply with the IEC60730 and IEC61508 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations. Two CRC functions are provided in the CMS32M65xx that can be used according to the application or purpose of use.

- “High-speed CRC” ... The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- “General CRC” ... This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(3) Frequency detection function

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the general-purpose timer unit.

(4) A/D test function

This is used to perform a self-check of the A/D converter by performing A/D conversion of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage.

(5) Digital output signal level detection function for input/output ports

When the input/output port is in output mode, the output level of the pin can be read.

27.2 Register mapping

The safety functions use the following registers for each function.

| Register name | Function |
|---------------------------------------------------|----------------------------------------------------------------------|
| • Flash CRC control register (CRC0CTL) | Flash CRC operation function (High-speed CRC) |
| • Flash CRC operation result register (PGCRCL) | |
| • CRC input register (CRCIN) | CRC calculation function (General CRC) |
| • CRC data register (CRCD) | |
| • Special SFR protection control register (SFRGD) | SFR guard function |
| • Timer input/output select register 0 (TIOS0) | Frequency detection function |
| • A/D test register (CON2) | A/D test function |
| • Port mode selection register (PMS) | Digital output signal level detection function for input/output pins |

The contents of each register are described in “27.3 Operation of safety function”.

(Flash memory CRC base address = 0x4002_1810)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------------------------|-------------|
| CRC0CTL | 0x000 | R/W | Flash memory CRC control register | 0x0 |
| PGCRCL | 0x002 | R/W | Flash memory CRC operation result register L | 0x0 |

(General CRC base address = 0x4004_32FA)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|--------------------------------------------|-------------|
| CRC0 | 0x000 | R/W | Flash memory CRC operation result register | 0x0 |
| CRCIN | 0x0B2 | R/W | Flash memory CRC control register | 0x0 |

(SFR base address = 0x4004_0478)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------|-------------|
| SFRGD | 0x000 | R/W | SFR guard control register | 0x0 |

(Port control base address = 0x4004_087B)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|---------------------------|-------------|
| PMS | 0x000 | R/W | Port mode select register | 0x0 |

(UID base address = 0x0050_0894)

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|--------------------------------|-------------|
| UID0 | 0x000 | RO | Product unique ID bit [31:0] | - |
| UID1 | 0X004 | RO | Product unique ID bit [63:32] | - |
| UID2 | 0X008 | RO | Product unique ID bit [95:64] | - |
| UID3 | 0X00C | RO | Product unique ID bit [127:96] | - |

27.3 Operation of safety functions

27.3.1 Flash CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC can be used to check the entire code flash memory area during the initialization routine.

The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 1024 μ s@64 MHz with 64-KB flash memory).

The CRC generator polynomial used complies with CRC-16-CCITT “ $X^{16}+X^{12}+X^5+1$ ”.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Note 1: This is the system clock, the relationship between the system clock and the clock of flash CRC operation is 4 : 1, the system clock is 64MHZ, then the clock of flash CRC operation is 16MHZ.

Note 2: The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range. The CRC0CTL register can be set by an 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

| Bit | Symbol | Description | Reset value |
|-----|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | CRC0EN | High-speed CRC operation control 0: Operation stopped 1: Start the operation by executing the WFE instruction | 0 |
| 6 | CRCCHK60 | 60K operation range selection 0: The range is controlled by bit[2:0]. 1: 00000H ~EFFFH(60K-4byte) | 0 |
| 5:3 | - | Reserved | - |
| 2:0 | FEA | High-speed CRC operation range 000: 00000H ~1FFFH(8K-4byte) 001: 00000H ~3FFFH(16K-4byte) 010: 00000H ~5FFFH(24K-4byte) 011: 00000H ~7FFFH(32K-4byte) 100: 00000H ~9FFFH(40K-4byte) 101: 00000H ~BFFFH(48K-4byte) 110: 00000H ~DFFFH(56K-4byte) 111: 00000H ~FFFFH(64K-4byte) | 0x0 |

Note 1: Bit3~5 must be set to 0.

Note 2: Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes

27.3.1.1 Flash memory CRC operation result register L (PGCRCL)

This register is used to store the lowest 16-bit results of the high-speed CRC operation.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

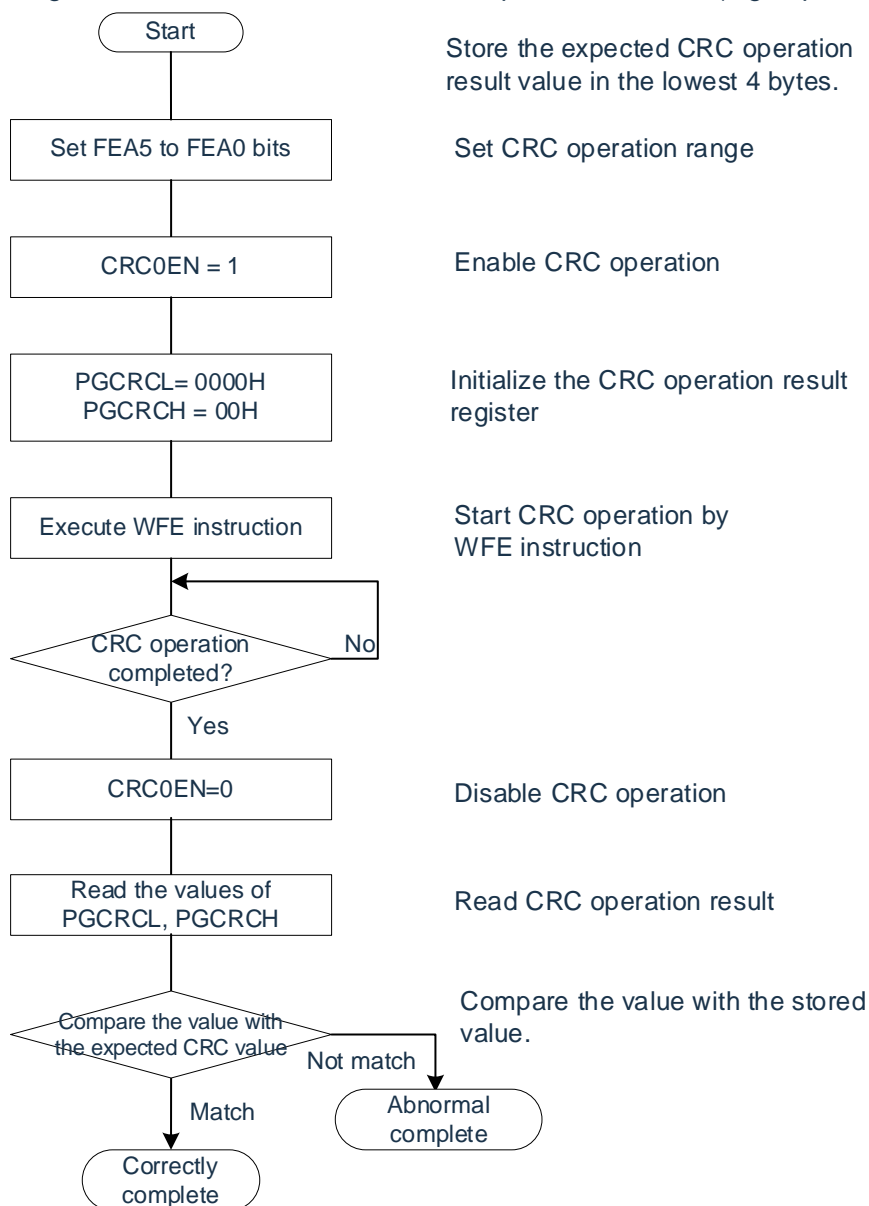
| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------|-------------|
| 15:0 | PGCRCL | Store the low 16-bit results of the high-speed CRC operation. 0000H ~FFFFH | 0x0 |

Note: The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

The flowchart of the flash memory CRC operation function (high-speed CRC) is shown in Figure 27-1.

<Operation flow>

Figure 27-1: Flow chart of flash CRC operation function (high-speed CRC)



Note 1: The CRC operation is executed only on the code flash.

Note 2: Store the expected CRC operation value in the area below the operation range in the code flash.

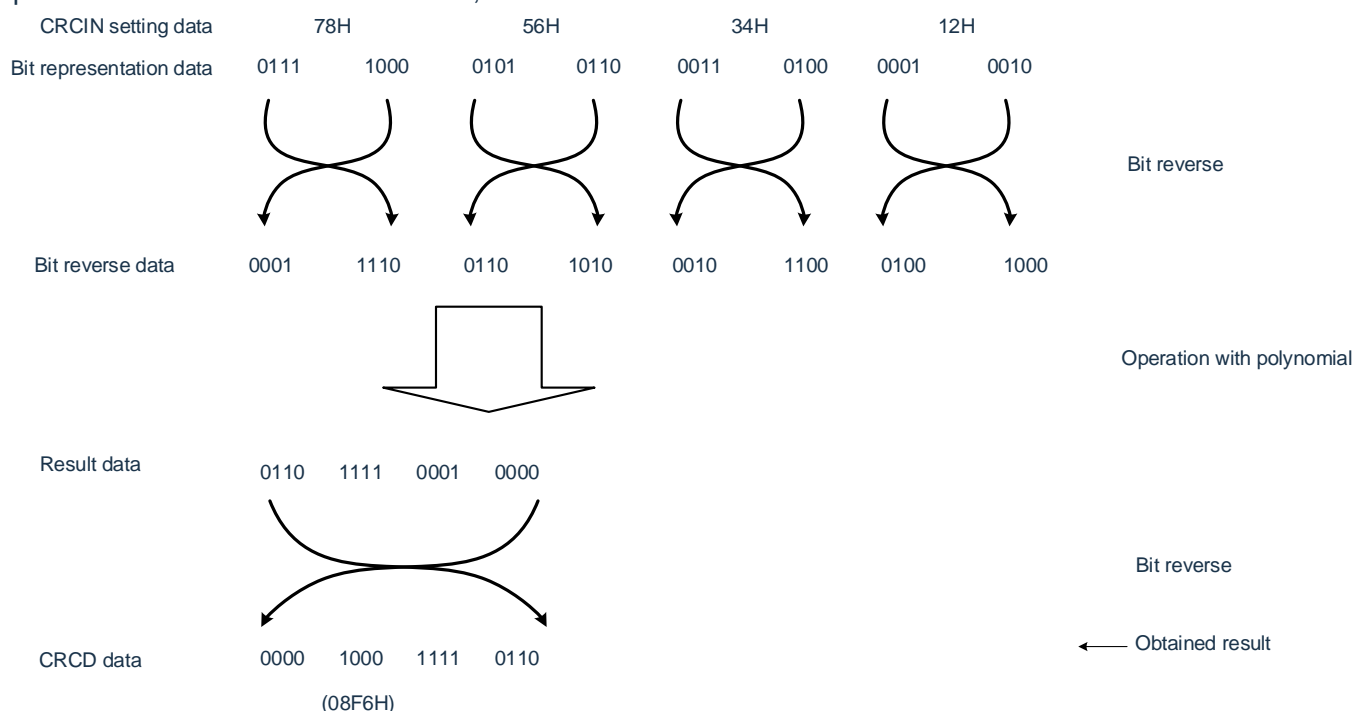
27.3.1.2 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

The general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program).

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

The CRC generator polynomial used is “ $X^{16}+X^{12}+X^5+1$ ” of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Note: Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

27.3.1.3 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC. The possible setting range is 00H to FFH.

The CRCIN register can be set by an 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------|-------------|
| 7:0 | CRCIN | General-purpose CRC data input 00H ~FFH | 0x0 |

27.3.1.4 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC. The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (F_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

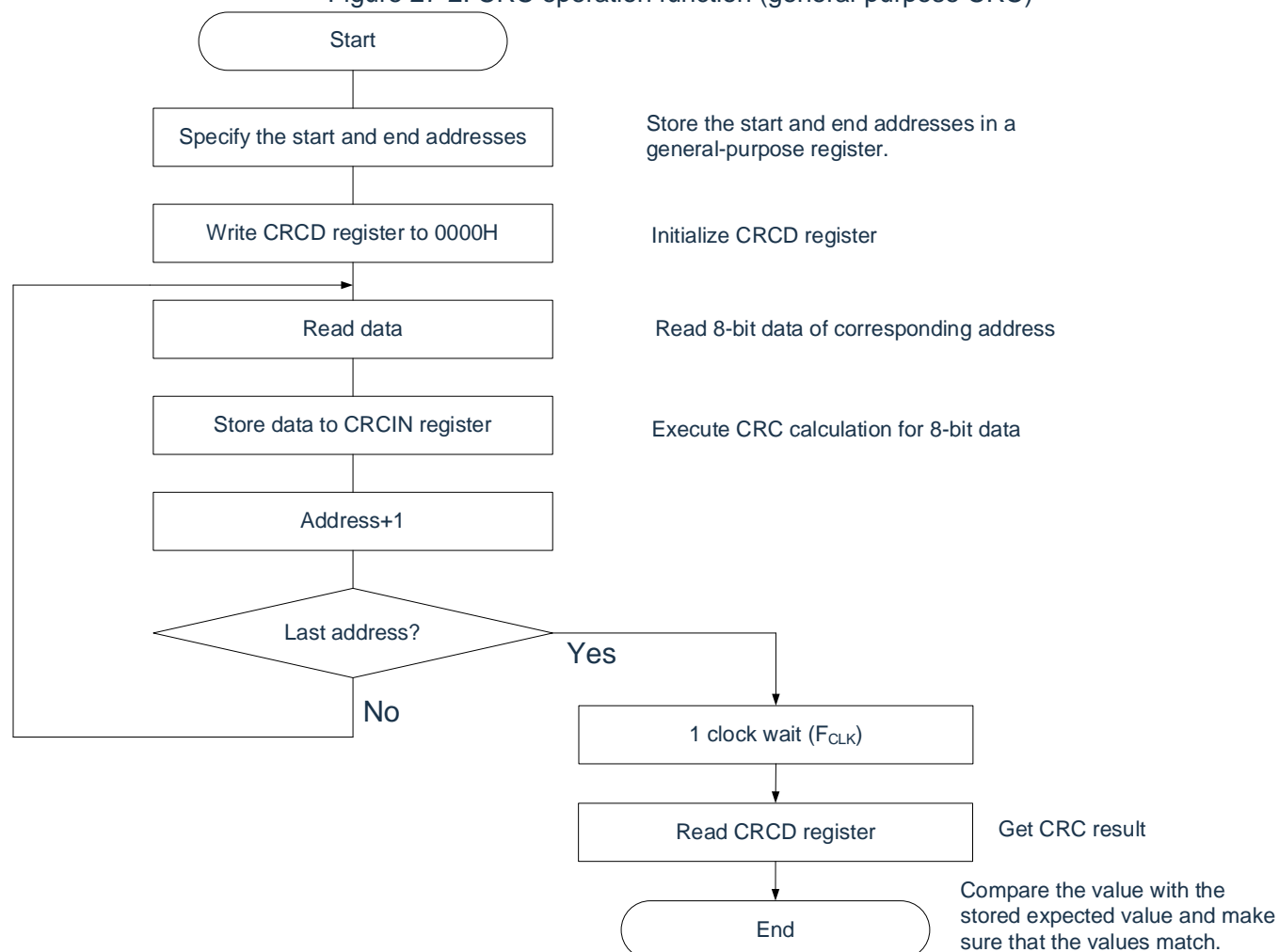
| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------|-------------|
| 15:0 | CRCD | Store the general-purpose CRC operation results 0000H ~FFFFH | 0x0 |

Note 1: To read the write value of the CRCD register, the CRCD register must be read before the CRCIN register is written.

Note 2: If a write operation to the CRCD register competes with the saving of an operation result, the write operation is ignored.

<Operation flow>

Figure 27-2: CRC operation function (general-purpose CRC)



27.3.2 SFR guard function

In order to ensure safety during operation, the IEC61508 standard requires that even if the CPU is out of control, it is necessary to protect important SFR from being rewritten. The SFR protection function is used to protect data from the control registers of the comparator function, port function, interrupt function, clock control function, and voltage detection circuitry.

If the SFR protection function is set, the write operation of the protected SFR is invalid, but it can be read normally.

27.3.2.1 SFR guard control register (SFRGD)

This register controls whether the SFR guard function is valid.

The SFR guard function uses GPORT bits and GCSC bits.

The SFRGD register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register becomes "00H".

| Bit | Symbol | Description | Reset value |
|-----|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | Reserved | - | - |
| 2 | GPORT | Protection of control registers for port functions 0: Invalid. Can read and write control registers for port functions. 1: Valid. Write operation of the control register of the port function is invalid, and it can be read. [Protected SFR] PMxx, PUxx, PDxx, POMxx, PMCxx, PxxCFG ^{Note} . | 0 |
| 1 | - | Reserved | - |
| 0 | GCSC | Clock control function, voltage detection circuit control register protection 0: Invalid. Can read/write clock control function, voltage detection circuit control register. 1: Valid. Clock control function and write operation of the control register of the voltage detection circuit are invalid and read operation is enabled. [Protected SFR] CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS. | 0 |

Note: Pxx (port register) is not protected.

27.3.3 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

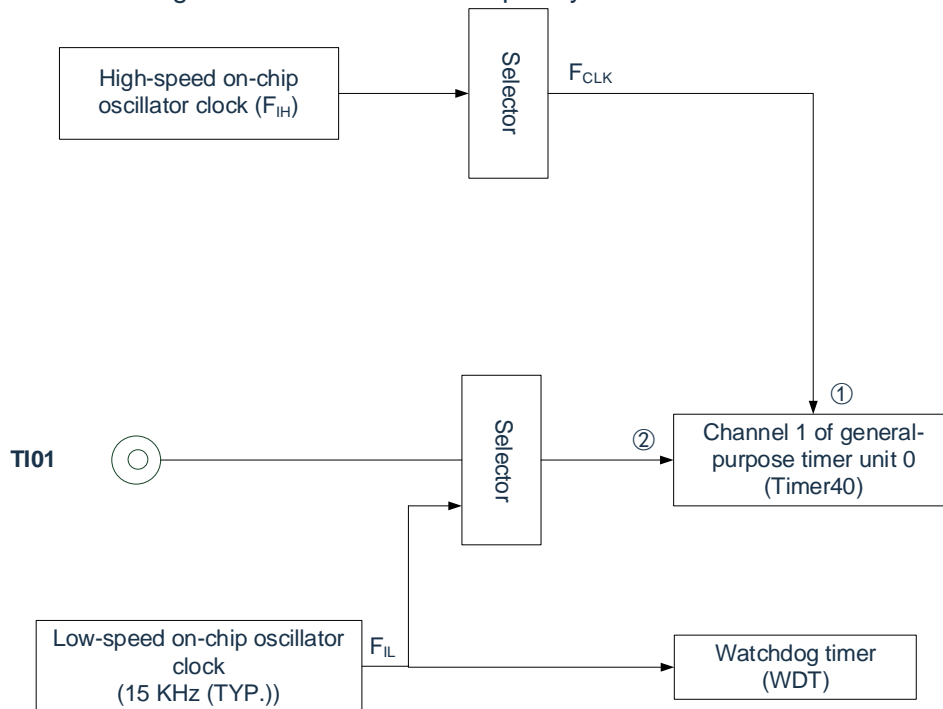
By using the CPU/peripheral hardware clock frequency (F_{CLK}) and measuring the pulse width of the input signal to channel 1 of the Timer40, whether the proportional relationship between the two clock frequencies is correct can be determined.

Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

- ① CPU/peripheral hardware clock frequency (F_{CLK}):
 - High-speed on-chip oscillator clock (F_{IH})
- ② Input to Channel 1 of the Timer40:
 - Timer input to channel 1 (TI01)
 - Low-speed on-chip oscillator clock (F_{IL} : 15kHz(TYP.))

Figure 27-3: Structure of frequency detection function



If the measurement result of the input pulse interval is an abnormal value, it can be judged as “clock frequency abnormality”. For the measurement method of the input pulse interval, refer to “6.7.4 Operation as input pulse interval measurement”.

Note: Can only be selected in the products incorporating the subsystem clock.

27.3.3.1 Timer input/output select register 0 (TIOS0)

Refer to Section 6.2.11 for the register description.

27.3.4 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANl), temperature sensor output voltage, and the internal reference voltage.

The analog multiplexer can be checked using the following procedure.

- ① Select the ANl pin for A/D conversion using the CON2 register (ADCSWCHS = 00100).
- ② Perform A/D conversion for the ANl pin (conversion result 1-1).
- ③ Select the A/D converter's negative reference voltage for A/D conversion using the CON2 register (ADCSWCHS = 10101)
 - ④ Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
 - ⑤ Select the ANl pin for A/D conversion using the CON2 register (ADCSWCHS = 00100).
 - ⑥ Perform A/D conversion for the ANl pin (conversion result 1-2).
 - ⑦ Select the A/D converter's positive reference voltage for A/D conversion using the CON2 register (ADCSWCHS = 10100)
 - ⑧ Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
 - ⑨ Select the ANl pin for A/D conversion using the CON2 register (ADCSWCHS = 00100).
 - ⑩ Perform A/D conversion for the ANl pin (conversion result 1-3).
 - ⑪ Check that the conversion results 1-1, 1-2, and 1-3 are equal.
 - ⑫ Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Note 1: If the analog input voltage is variable during A/D conversion in steps ①~⑩ above, use another method to check the analog multiplexer

Note 2: The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

27.3.4.1 ADC register

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channel (ANlxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select positive reference voltage as the target of A/D conversion for full-scale measurement.

Note 1: When using this mode, the positive (+) reference voltage of the A/D should be selected as VDD.

Note 2: To use this mode, ADCSWCHE needs to be set to 1 (channel is controlled by ADCSWCHS).

Please refer to Chapter 17 for A/D registers and instructions.

27.3.5 Digital output signal level detection function for input/output pins

The IEC60730 standard mandates confirming that the I/O functions are normal.

Input/Output Pin Digital Output Signal Level Detection Function reads the digital output level of a pin when the pin is in output mode.

27.3.5.1 Port mode select register (PMS)

This register selects whether to read the value of the port's output latch or the output level of the pin when the pin is in output mode (PMmn bit of the Port Mode Register (PMm) is "0").

The PMS register is set by an 8-bit memory manipulation instruction.

After a reset signal is generated, the value of this register changes to "00H".

| Bit | Symbol | Description | Reset value |
|-----|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:1 | - | Reserved | - |
| 0 | PMS0 | Selection of reading data when the pin is in output mode 0: Reads the value of the Pmn register. 1: Read the digital output level of the pin. | 0 |

Note 1: If the digital output level of the pin is read, the read value is "0" for the pin that has been changed to a high impedance state by using the pulse output forced truncation function of Timer M.

Note 2: m=0~2

n=0~7

27.3.6 Product unique ID register

The unique ID of the product is perfect for:

- (1) Used as a serial number (e.g. USB character serial number or other terminal applications).
- (2) Used as a password, this unique ID is used in conjunction with a software encryption and decryption algorithm when writing flash memory to improve the security of the code in the flash memory.
- (3) Used to activate a bootstrap process with a safety mechanism

The reference number provided by the 128-bit product unique ID is unique to any microcontroller in any case. Under any circumstances, the user cannot modify this ID.

Product unique ID register 0 (UID0)

| Bit | Symbol | Description | Reset value |
|------|--------|-----------------------------------------------------------------------------------------|-------------|
| 31:0 | - | Product unique ID register bit [31:0], the value of which is programmed at the factory. | - |

Product unique ID register 1 (UID1)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------|-------------|
| 31:0 | - | Product unique ID register bit [63:32] , the value of which is programmed at the factory. | - |

Product unique ID register 2 (UID2)

| Bit | Symbol | Description | Reset value |
|------|--------|-------------------------------------------------------------------------------------------|-------------|
| 31:0 | - | Product unique ID register bit [95:64] , the value of which is programmed at the factory. | - |

Product unique ID register 3 (UID3)

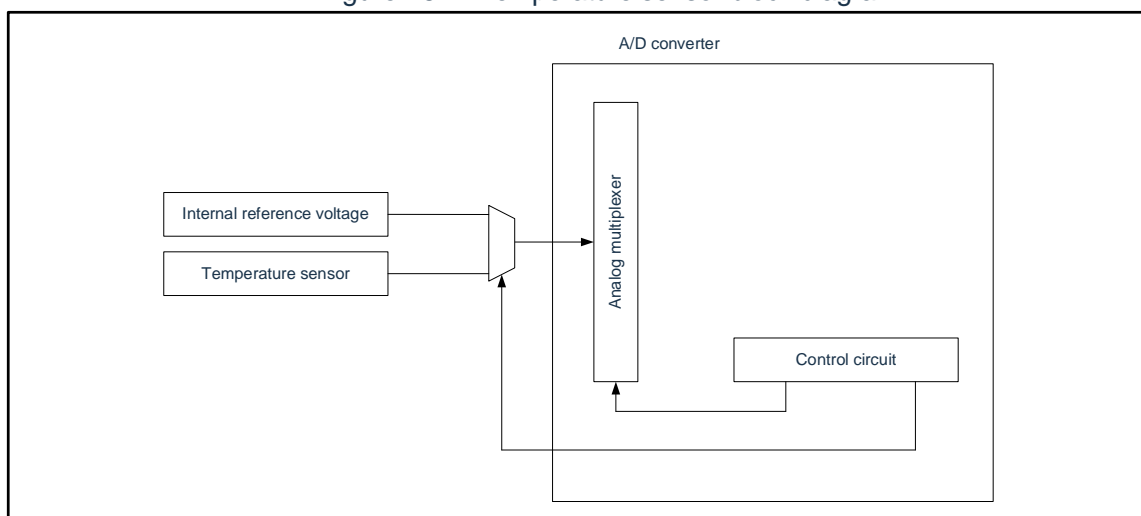
| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------|-------------|
| 31:0 | - | Product unique ID register bit [127:96] , the value of which is programmed at the factory. | - |

Chapter 28 Temperature Sensor

28.1 Function of temperature sensor

The on-chip temperature sensor measures and monitors the core temperature of the product, thus ensuring reliable operation of the product. The voltage output by the temperature sensor is proportional to the core temperature, and there is a linear relationship between the voltage and temperature. Its output voltage is supplied to the ADC for conversion. Figure 28-1 shows a block diagram of a temperature sensor.

Figure 28-1: Temperature sensor block diagram



28.2 Register mapping

RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Address | R/W | Description | Reset value |
|----------|------------|-----|----------------------------------------------|-------------|
| TSN25 | 0x0050066C | RO | Temperature Sensor Calibration Data Register | - |

28.3 Temperature sensor register

28.3.1 Temperature sensor calibration data register TSN25

| Bit | Symbol | Description | Reset value |
|-------|--------|--------------------------------------------------------------------------------------------------------------------|-------------|
| 15:12 | - | Reserved | - |
| 11:0 | TSN25 | Calibration data 1, automatically loaded at power on or reset startup, and each chip has its own calibration data. | - |

28.4 Instructions for using temperature sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so the temperature is calculated as follows:

$$T = (V_s - V_1) / \text{slope} + 25^{\circ}\text{C}$$

T: Measured temperature (°C)

Vs: Output voltage of the temperature sensor at temperature measurement (V)

V1: Voltage output at 25°C measured by temperature sensor (V)

Slope: Temperature slope of the temperature sensors(V/°C), slope = -3.5 mV/°C

Remark: Temperature sensors have low accuracy and are not recommended for use in applications where high accuracy is required.

Chapter 29 Option Byte

29.1 Function of option byte

Addresses 000C0H~000C3H, 500004H of the flash emory of the CMS32M65xx form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and Flash memory data protection option byte (000C3H, 500004H). When powered on or reset is initiated, the specified function is set with reference to the option byte. When using this product, the following functions must be set by the option byte. For bits that do not have configuration capabilities, you cannot change the initial value.

Caution: Regardless of whether or not to use each function, you must set the option byte.

29.1.1 User option bytes (000C0H~000C2H)

(1) 000C0H

- Operation of watchdog timer
 - Enable or disable counter operation.
 - Enable or stop counter operation in sleep/deep sleep mode.
- Setting of watchdog timer overflow time
 - Setting of window open period of watchdog timer
- Setting of interval interrupt of watchdog timer
 - Whether or not to use the interval interrupt is selectable.

(2) 000C1H

- Setting of LVD operation mode
 - Interrupt & reset mode.
 - Reset mode.
 - Interrupt mode.
 - LVD off (by controlling the externally input reset signal on the RESETB pin)
- Setting of LVD detection level (V_{LVDH} , V_{LVDL} , V_{LVD})

Note: When the supply voltage rises, the reset state must be maintained by voltage detection circuits or external resets before the supply voltage reaches the operating voltage range shown in the AC characteristics of the data sheet; When the supply voltage drops, it must be reset by transferring in deep sleep mode, voltage detection circuitry, or external reset before the supply voltage falls below the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H).

(3) 000C2H

- Setting of the frequency of the high-speed on-chip oscillator
 - Select from 2MHz~32MHz, 48MHz, 64MHz.

29.2 Flash memory data protection option bytes (000C3H, 500004H)

- Control of flash memory data protection when debugging on-chip

Level0: Read/write/erase operations on flash data are enabled via debugger.

Level1: Chip erase operations on flash data via debugger are enabled, read/write operations are disabled.

Level2: Operations on flash data via debugger are disabled.

29.3 Register mapping

| Register | Address | R/W | Description | Reset value |
|---------------|-----------|-----|----------------------------------------------------------|-------------|
| Option byte 0 | 0x00C0H | R/W | Watchdog Timer Status Control Register | 0xFF |
| Option byte 1 | 0x00C1H | R/W | LVD Status Control Register | 0xFF |
| Option byte 2 | 0x00C2H | R/W | High-Speed On-Chip Oscillator Frequency Control Register | 0xEC |
| Option byte 3 | 0x00C3H | R/W | On-chip Debug Flash Data Protection Control Register 1 | 0xFF |
| Option byte 4 | 0x500004H | R/W | On-chip Debug Flash Data Protection Control Register 2 | 0xFF |

29.4 User option bytes

29.4.1 User option byte (000C0H)

| Bit | Symbol | Description | Reset value |
|-----|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7 | WDTINT | Interval interrupt of watchdog timer 0: Interval interrupt is not used. When 75% of the overflow time + 1: 1/2F _{IL} is reached, an interval interrupt is generated. | 1 |
| 6:5 | WINDOW[1:0] | When watchdog timer window opens 0X: Settings are disabled. 10: 75% 11: 100% | 0x3 |
| 4 | WDTON | Controlling counter operation of watchdog timer 0: Disable counter operation (stop counting after the reset is released). 1: Enable counter operation (start counting after the reset is released). | 1 |
| 3:1 | WDTCS[2:0] | Overflow time of watchdog timer (F _{IL} =20kHz (MAX.)) 000: 2 ⁶ /F _{IL} (3.2ms) 001: 2 ⁷ /F _{IL} (6.4ms) 010: 2 ⁸ /F _{IL} (12.8ms) 011: 2 ⁹ /F _{IL} (25.6ms) 100: 2 ¹¹ /F _{IL} (102.4ms) 101: 2 ¹³ /F _{IL} (409.6ms) 110: 2 ¹⁴ /F _{IL} (819.2ms) 111: 2 ¹⁶ /F _{IL} (3276.8ms) | 0x7 |
| 0 | WDSTBYON | Counter operation control (sleep mode) of watchdog timer 0: In sleep mode, counter operations are stopped ^{Note1} . 1: In sleep mode, counter operations are enabled. | 1 |

Note 1: When the WDSTBYON bit is “0”, regardless of the values of the WINDOW1 bit and the WINDOW0 bit, it is 100% during window opening.

Note 2: F_{IL}: Low-speed on-chip oscillator clock frequency

29.4.2 User option byte (000C1H)

| Bit | Symbol | Description | Reset value |
|-----|-------------|----------------------------|-------------|
| 7:5 | VPOC[2:0] | Detection voltage setting | 0x7 |
| 4 | - | Reserved (Must be 1) | 1 |
| 3:2 | LVIS[1:0] | Detection voltage setting | 0x2 |
| 1:0 | LVIMDS[1:0] | Mode selection | 0x3 |
| | | 10: Interrupt & reset mode | |
| | | 11: Reset mode | |
| | | 01: Interrupt mode | |

LVD settings (interrupt & reset mode)

| Detect voltage | | | Setting value of option byte | | | | | | |
|----------------|---------|---------|-------------------------------------------|-------|-------|-------|-------|--------------|---------|
| VLVDH | | VLVDL | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 | Mode setting | |
| Rising | Falling | Falling | | | | | | LVIMDS1 | LVIMDS0 |
| 1.77V | 1.73V | 1.63V | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1.88V | 1.84V | | | | | 0 | 1 | | |
| 2.92V | 2.86V | | | | | 0 | 0 | | |
| 1.98V | 1.94V | 1.84V | | 0 | 1 | 1 | 0 | | |
| 2.09V | 2.04V | | | | | 0 | 1 | | |
| 3.13V | 3.06V | | | | | 0 | 0 | | |
| 2.61V | 2.55V | 2.45V | | 1 | 0 | 1 | 0 | | |
| 2.71V | 2.65V | | | | | 0 | 1 | | |
| 3.75V | 3.67V | | | | | 0 | 0 | | |
| 2.92V | 2.86V | 2.75V | | 1 | 1 | 1 | 0 | | |
| 3.02V | 2.96V | | | | | 0 | 1 | | |
| 4.06V | 3.98V | | | | | 0 | 0 | | |
| — | | | Settings other than above are prohibited. | | | | | | |

Note 1: Bit4 must be written as “1”.

Note 2: For details of LVD circuit, please refer to “Chapter 26 Voltage Detection Circuit”.

Note 3: The detection voltage is a TYP value. For details, please refer to the LVD circuit characteristics in the data sheet.

LVD setting (reset mode)

| Detect voltage | | Setting value of option byte | | | | | | |
|----------------|---------|-------------------------------------------|-------|-------|-------|-------|--------------|---------|
| VLVD | | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 | Mode setting | |
| Rising | Falling | | | | | | LVIMDS1 | LVIMDS0 |
| 1.67V | 1.63V | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1.77V | 1.73V | | 0 | 0 | 1 | 0 | | |
| 1.88V | 1.84V | | 0 | 1 | 1 | 1 | | |
| 1.98V | 1.94V | | 0 | 1 | 1 | 0 | | |
| 2.09V | 2.04V | | 0 | 1 | 0 | 1 | | |
| 2.50V | 2.45V | | 1 | 0 | 1 | 1 | | |
| 2.61V | 2.55V | | 1 | 0 | 1 | 0 | | |
| 2.71V | 2.65V | | 1 | 0 | 0 | 1 | | |
| 2.81V | 2.75V | | 1 | 1 | 1 | 1 | | |
| 2.92V | 2.86V | | 1 | 1 | 1 | 0 | | |
| 3.02V | 2.96V | | 1 | 1 | 0 | 1 | | |
| 3.13V | 3.06V | | 0 | 1 | 0 | 0 | | |
| 3.75V | 3.67V | | 1 | 0 | 0 | 0 | | |
| 4.06V | 3.98V | | 1 | 1 | 0 | 0 | | |
| — | | Settings other than above are prohibited. | | | | | | |

Note 1: Bit4 must be written as “1”.

Note 2: For details of LVD circuit, please refer to “Chapter 26 Voltage Detection Circuit”.

Note 3: The detection voltage is a TYP value. For details, please refer to the LVD circuit characteristics in the data sheet.

LVD setting (interrupt mode)

| Detect voltage | | Setting value of option byte | | | | | | |
|----------------|---------|-------------------------------------------|-------|-------|-------|-------|--------------|---------|
| VLVD | | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 | Mode setting | |
| Rising | Falling | | | | | | LVIMDS1 | LVIMDS0 |
| 1.67V | 1.63V | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1.77V | 1.73V | | 0 | 0 | 1 | 0 | | |
| 1.88V | 1.84V | | 0 | 1 | 1 | 1 | | |
| 1.98V | 1.94V | | 0 | 1 | 1 | 0 | | |
| 2.09V | 2.04V | | 0 | 1 | 0 | 1 | | |
| 2.50V | 2.45V | | 1 | 0 | 1 | 1 | | |
| 2.61V | 2.55V | | 1 | 0 | 1 | 0 | | |
| 2.71V | 2.65V | | 1 | 0 | 0 | 1 | | |
| 2.81V | 2.75V | | 1 | 1 | 1 | 1 | | |
| 2.92V | 2.86V | | 1 | 1 | 1 | 0 | | |
| 3.02V | 2.96V | | 1 | 1 | 0 | 1 | | |
| 3.13V | 3.06V | | 0 | 1 | 0 | 0 | | |
| 3.75V | 3.67V | | 1 | 0 | 0 | 0 | | |
| 4.06V | 3.98V | | 1 | 1 | 0 | 0 | | |
| — | | Settings other than above are prohibited. | | | | | | |

Note 1: Bit4 must be written as “1”.

Note 2: For details of LVD circuit, please refer to “Chapter 26 Voltage Detection Circuit”.

Note 3: The detection voltage is a TYP value. For details, please refer to the LVD circuit characteristics in the data sheet.

Setting when LVD is OFF (external reset input using RESETB pin)

| Detect voltage | | Setting value of option byte | | | | | | |
|----------------|---------|-------------------------------------------|-------|-------|-------|-------|--------------|---------|
| VLVDH | | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 | Mode setting | |
| Rising | Falling | | | | | | LVIMDS1 | LVIMDS0 |
| — | — | 1 | × | × | × | × | × | 1 |
| — | | Settings other than above are prohibited. | | | | | | |

Note 1: Bit4 must be written as “1”.

Note 2: When the supply voltage rises, the reset state must be maintained by the voltage detection circuit or external reset before the supply voltage reaches the operating voltage range shown in the AC Characteristics of the datasheet; when the supply voltage falls, the reset state must be reset by the transfer of the sleep mode, the voltage detection circuit, or the external reset before the supply voltage falls below the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H).

Note 3: x: Ignore

Note 4: For details of LVD circuit, please refer to “Chapter 26 Voltage Detection Circuit”.

Note 5: The detection voltage is a TYP value. For details, please refer to the LVD circuit characteristics in the data sheet.

29.4.3 User option byte (000C2H)

| Bit | Symbol | Description | Reset value |
|-----|------------|---------------------------------------------------------|-------------|
| 7:5 | - | Reserved (Must be 1) | 0x7 |
| 4:0 | FRQSE[4:0] | High-speed on-chip oscillator clock frequency selection | 0x0C |

| FRQSE4 | FRQSEL3 | FRQSEL2 | FRQSEL1 | FRQSEL0 | High-speed on-chip oscillator clock frequency | |
|----------------------|---------|---------|---------|---------|-----------------------------------------------|-----------------|
| | | | | | F _{HOCO} | F _{IH} |
| 0 | 1 | 0 | 0 | 0 | 64MHz | 64MHz |
| 0 | 0 | 0 | 0 | 0 | 48MHz | 48MHz |
| 0 | 1 | 0 | 0 | 1 | 64MHz | 32MHz |
| 0 | 0 | 0 | 0 | 1 | 48MHz | 24MHz |
| 0 | 1 | 0 | 1 | 0 | 64MHz | 16MHz |
| 0 | 0 | 0 | 1 | 0 | 48MHz | 12MHz |
| 0 | 1 | 0 | 1 | 1 | 64MHz | 8MHz |
| 0 | 0 | 0 | 1 | 1 | 48MHz | 6MHz |
| 0 | 1 | 1 | 0 | 0 | 64MHz | 4MHz |
| 0 | 0 | 1 | 0 | 0 | 48MHz | 3MHz |
| 0 | 1 | 1 | 0 | 1 | 64MHz | 2MHz |
| Other than the above | | | | | Settings are prohibited | |

Note 1: Bits 7 to 5 must be set to “1”.

Note 2: Operating frequency range and operating voltage range vary depending on each operating mode of the flash memory. For details, refer to AC Characteristics in the datasheet.

29.4.4 Flash memory data protection option byte (000C3H)

| Bit | Symbol | Description | Reset value |
|-----|------------|-----------------------------------------|-------------|
| 7:0 | OCDEN[7:0] | Control of flash memory data protection | 0xFF |

29.4.5 Flash memory data protection option byte (500004H)

| Bit | Symbol | Description | Reset value |
|-----|-----------|-----------------------------------------|-------------|
| 7:0 | OCDM[7:0] | Control of flash memory data protection | 0xFF |

| OCDM | OCDEN | Description |
|----------------------|-------|-----------------------------------------------------------------------------------------------|
| 3C | C3 | Manipulation of flash data via debugger is disabled. |
| Other than 3C | C3 | Chip erase operation on flash data via debugger is enabled, read/write operation is disabled. |
| Other than the above | | Read/write/erase operations on flash data via debugger are enabled. |

Note: The 50_0004H address belongs to the data flash memory area. If you use this address for data storage, make sure that the value will not cause the protection option to be set incorrectly.

Chapter 30 FLASH Control

30.1 Overview of FLASH control

This product contains a 64KByte FLASH memory, which is divided into 128 Sectors, each with a capacity of 512 Bytes. It can be used as program memory and data memory. This module supports erase, program and read operations for this memory.

30.2 Structure of FLASH memory

| | |
|-------------|-----------------------------------------------|
| FFFF_FFFFH | Reserved |
| E00F_FFFFH | Cortex-M0+ Dedicated Peripheral Resource Area |
| E000_0000H | |
| | Reserved |
| 4006_FFFFH | Peripheral Resource Area |
| | |
| | |
| 4000_0000H | Reserved |
| 2000_1FFFFH | SRAM (up to 8KB) |
| 2000_0000H | |
| | Reserved |
| 0050_05FFFH | Data Flash 1KB |
| 0050_0200H | |
| | Reserved |
| 0000_FFFFH | Main Flash Memory Area (up to 64KB) |
| | |
| 0000_0000H | |

30.3 Register mapping

(FLASH control base address = 0x4002_0000) RO: Read only, WO: Write Only, R/W: Read/Write

| Register | Offset value | R/W | Description | Reset value |
|----------|--------------|-----|----------------------------------------|-------------|
| FLSTS | 0x000 | R/W | FLASH Status Register | 0x0 |
| FLOPMD1 | 0x004 | R/W | FLASH Operation Control Register 1 | 0x0 |
| FLOPMD2 | 0x008 | R/W | FLASH Operation Control Register 2 | 0x0 |
| FLERMD | 0x00C | R/W | FLASH Erase Control Register | 0x0 |
| FLCERCNT | 0x010 | R/W | FLASH Chip Erase Time Control Register | - |
| FLSERCNT | 0x014 | R/W | FLASH Page Erase Time Control Register | - |
| FLPROCNT | 0x01C | R/W | FLASH Write Time Control Register | - |
| FLPROT | 0x020 | R/W | FLASH Write Protect Register | 0x0 |

30.4 Register description

30.4.1 Flash write protection register (FLPROT)

Flash protection register is a register used to protect the FLASH operation control register.

| Bit | Symbol | Description | Reset value |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:1 | PRKEY | WRP write protection 78h: Enable rewriting WRP Other: Disable rewriting WRP | 0x0 |
| 0 | WRP | Operation register (FLOPMD1/FLOPMD2) write-protection 0: Disable rewriting FLOPMD1/FLOPMD2 1: Enable rewriting FLOPMD1/FLOPMD2 | 0 |

30.4.2 FLASH operation control register (FLOPMD1)

Flash operation control register is used to set the erase and write operations of FLASH.

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | FLOPMD1 | FLASH operation select bit: 0x55: When FLOPMD2=0xAA: Erase 0xAA: When FLOPMD2=0x55: Write 0x00: When FLOPMD2=0x00: Read Other than the above: Settings are disabled | 0x0 |

30.4.3 FLASH operation control register (FLOPMD2)

Flash operation control register is used to set the erase and write operations of FLASH.

| Bit | Symbol | Description | Reset value |
|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 31:8 | - | Reserved | - |
| 7:0 | FLOPMD2 | FLASH operation select bit: 0xAA: When FLOPMD1=0x55: Erase 0x55: When FLOPMD1=0xAA: Write 0x00: When FLOPMD1=0x00: Read Other than the above: Settings are disabled | 0x0 |

30.4.4 Flash erase control register (FLERMD)

Flash erase control register is used to set the type of FLASH erase operation.

| Bit | Symbol | Description | Reset value |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:5 | - | Reserved | - |
| 4:3 | ERMD | Erase operation control bit. 0: Sector erase, no hardware check after erase 1: Chip erase ^{Note} 2: Sector erase, hardware check after erase 3: Settings are disabled | 0x0 |
| 2:0 | - | Reserved | - |

Note: Chip erase only erases the code flash area, not the data flash area. And chip erase does not support hardware check.

30.4.5 Flash status register (FLSTS)

The status of the FLASH controller can be queried through the status register.

| Bit | Symbol | Description | Reset value |
|-----|--------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 7:3 | - | Reserved | - |
| 2 | EVF | FLASH erase hardware check error flag 0: No hardware check error after FLASH erase 1: Hardware check error occurs after FLASH erase | 0 |
| 1 | - | Reserved | - |
| 0 | OVF | FLASH erase/write operation complete flag 0: FLASH erase/write operation is not completed 1: FLASH erase/write operation is completed | 0 |

Note 1: The OVF needs to be cleared by writing “1” through software. If it is not cleared, the next erase operation cannot be performed.

Note 2: The EVF needs to be cleared by writing “1” through software.

30.4.6 Flash chip erase time control register (FLCERCNT)

FLCERCNT register enables to set the FLASH chip erase time.

| Bit | Symbol | Description | Reset value |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 31 | Load | Chip erase time setting selection ^{Note} 0: Erase time is set by hardware 1: Erase time is set by software (FLCERCNT[9:0]) | 0 |
| 30:10 | - | Reserved | - |
| 9:0 | FLCERCNT | Software erase time setting Chip erase time = (FLCERCNT*2048*Tfclk), which meets the hardware requirement of >30ms | Undefined value |

Note: When the master clock is an on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, the hardware setting time can be used without setting FLCERCNT.

30.4.7 Flash sector erase time control register (FLSERCNT)

LSERCNT register enables to set the FLASH sector erase time.

| Bit | Symbol | Description | Reset value |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 31 | Load | Sector erase time setting selection ^{Note} 0: Erase time is set by hardware 1: Erase time is set by software (FLSERCNT[10:0]) | 0 |
| 30:11 | - | Reserved | - |
| 10:0 | FLSERCNT | Software erase time setting sector erase time = (FLSERCNT*256*Tfclk) which meets the hardware requirement of >2ms | Undefined value |

Note: When the master clock is an on-chip high-speed OCO or the external input clock is $\leq 20\text{M}$, the time can be set in hardware without setting FLSERCNT.

30.4.8 Flash write time control register (FLPROCNT)

FLPROCNT register enables to set the FLASH WORD write time.

| Bit | Symbol | Description | Reset value |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 31 | Load1 | Write action setup time (T_{PGS}) setting ^{Note 1} 0: Write action setup time by hardware 1: Write action setup time by software FLPGSCNT[12:0] | 0 |
| 30:29 | - | Reserved | - |
| 28:16 | FLPGSCNT | Write action setup time by software Write action setup time = (FLPGSCNT * T_{fclk}), which meets the hardware requirement of > 70us for a word and > 40us for half words. | Undefined value |
| 15 | Load0 | Write time (T_{PROG}) setting selection ^{Note 2} 0: Write time is set by hardware 1: Write time is set by software FLPROCNT[8:0] | 0 |
| 14:9 | - | Reserved | - |
| 8:0 | FLPROCNT | Software write time setting Write time = (FLPROCNT * T_{fclk}), which meets the hardware requirement of > 7us | Undefined value |

Note 1: When the master clock is an on-chip high-speed OCO or the external input clock is $\leq 20M$, you can set the time by hardware without setting FLPGSCNT.

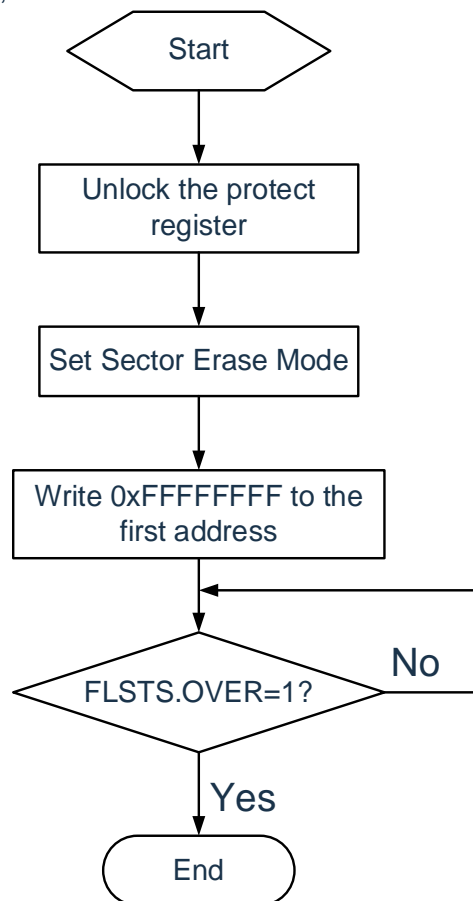
Note 2: When the master clock is an on-chip high-speed OCO or the external input clock is $\leq 20M$, you can set the time by hardware without setting FLPROCNT.

30.5 How to operate FLASH

30.5.1 Sector erase

The Sector erase time is realized by the hardware or can be configured by FLSERCNT. The operation flow is as follows.

- 1) Set FLERMD.ERMD0 to 1'b0, select the sector erase mode, and set the value of ERMD1 according to whether or not hardware check is required.
- 2) Set FLPROT to 0xF1, unprotect FLOPMD. Then set FLOPMD1 to 0x55, FLOPMD2 to 0xAA, and
- 3) Write arbitrary data to the first address of the erase target sector. Example: * ((unsigned long *)0x00000200)=0xffffffff.
- 4) Query the status register FLSTS.OVF through software, when OVF=1, it means the erase operation is completed.
- 5) If hardware check after erase is set (ERMD1=1), you can determine whether the verification is correct by software for FLSTS.EVF.
- 6) Before the next operation, set the software to "1" to clear the FLSTS.



30.5.2 Chip erase

Chip erase, and the erase time are implemented by hardware and can also be configured via FLCERCNT. The operation process is as follows

- 1) Set FLERMD. ERMD0 to 1'b1, and select chip erase mode;
- 2) Set FLPROT to 0xF1 to unprotect FLOPMD. Then set FLOPMD1 to 0x55 and FLOPMD2 to 0xAA.
- 3) Write arbitrary data to any address in the flash area of the code.
- 4) Query the status register FLSTS.OVF through software, when OVF=1, it means the erase operation is completed.
- 5) Before the next operation, set the software to “1” to clear the FLSTS.

30.5.3 Word program

Word programming and write time are implemented by hardware and can also be configured via PROCNT. The operation process is as follows:

- 1) Set FLPROT to 0xF1, unprotect FLOPMD. Then set FLOPMD1 to 0xAA, FLOPMD2 to 0x55, and
- 2) Write the corresponding data to the target address.
- 3) Query the status register FLSTS.OVF through software, when OVF=1, it means the write operation is completed.
- 4) Before the next operation, set the software to “1” to clear the FLSTS.

30.6 Flash memory read

The fastest fetch frequency supported by the built-in FLASH is 32 MHz. when the HCLK frequency exceeds 32 MHz, the hardware will insert 1 wait cycle when the CPU accesses the FLASH.

30.7 Cautions for FLASH operation

- 1) Flash memory has strict time requirements for the control signal of erasing and programming operation, and the timing of the control signal is not qualified, which will cause the erase operation and programming operation to fail. The setting of the erase and write parameters can be implemented by hardware, or it can be modified by modifying the parameter registers; When using on-chip high-speed OCO, MAINOSC/ external input clock = 16M, it is recommended to use hardware-set erase and write parameters without setting parameter registers.
- 2) If the erase/write operation is executed from FLASH, the CPU stops fetching and the hardware automatically waits for the completion of the operation to proceed to the next instruction. If the operation is executed from RAM, the CPU will not stop fetching and can continue the next instruction.
- 3) If the CPU executes an instruction to enter deep sleep while the FLASH is in programming operation, the system waits for the programming action to end before entering deep sleep.

Appendix Revision History

| Version | Date | Revision decription |
|---------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V0.1.0 | February 2023 | Initial version |
| V0.5.0 | August 2023 | <ol style="list-style-type: none"> 1) Updated formatting. 2) Corrected the referenced links in the full text. 3) Modified registers in section 3.4.9 and updated description in Table 3-2. 4) Corrected symbol SELLOSC in sections 5.4.6 and 5.7. 5) Added a subheading 6.1. 6) Corrected some errors in sections 1.2, 2.1, 3.2, 5.4.3, 6.2.4, 7.4, 8.3.2, 12.3.1, 14.1, 17.1, 17.2, 19.5, 20.3. 7) Added a END register in sections 14.4 and 14.5.9. 8) Modified content in sections 17.3.1, 17.3.2, 17.3.6 and 17.5.1. 9) Added explanation to Figure 17-1: ADC hardware trigger start in section 17.3.8. 10) Modified contents in section 17.4 and removd the register DATA19. 11) Modified register contents in sections 17.5.2 and 17.5.4, and added "Note". 12) Added explanation for channel enable register in section 17.5.5. 13) Added conditions or remarks in sections 17.5.5~17.5.14. 14) Removed section 17.5.9 ADC Test Register (TEST). 15) Deleted content related to LPF. 16) Corrected register name in section 20.5.2. 17) Modified structure diagram in section 21.2. 18) Added relevant description for PGCRCH register in Chapter 27 Safety Functions. 19) Modified content in section 27.3.4.1 ADC register. 20) Corrected register options byte 2 and bit[4:0] reset values in section 29.4.3. 21) Corrected some contents in sections 30.4.1, 30.4.6~30.4.8. 22) Deleted RAM1 and modified RAM0 name. 23) Removed temperature sensor usage method 1 in section 28.3 and only provided ambient temperature data. 24) Removed WDTE bit reset function descriptions in Chapter 9. 25) Modified ADC diagram in section 17.3.2, changing ADGO to ADCST. 26) Updated description of ADC control register in section 17.5.1. 27) Corrected register name for user option byte register in section 29.4.1. 28) Corrected description in section 6.5.3, changing from Figure 6-28 to Figure 6-14. 29) Corrected some content in Figure 6-43 (d) and (e) of the section 6.8.3. 30) Corrected contents of register LVIM in section 26.4.1 and register LVIS in section 26.4.2. |
| V0.9.0 | October 2023 | <ol style="list-style-type: none"> 1) Added cautions for END register in section 14.5.9. 2) Revised PMCxx register descriptions 3) Changed TAU0_TINO-3 to TI00, TI01, TI02, and TI03 in section 3.4.10 Port input multiplexing configuration register (PSxx_CFG). |

| | | |
|--------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | 4) Corrected the timing diagram for 12-bit interval timer in Figure 7-1. 5) Updated NIVC to support discrete interrupt numbers. 6) Revised LVIM register description. 7) Modified some contents in section 6.2.11. 8) Modified some contents in Chapter 27. 9) Modified some contents in section 28.2. |
| V0.9.1 | October 2023 | Added description for the BG channel. |
| | November 2023 | Modified some contents in section 27.3.6 |
| | January 2024 | 1) Added section 9.3.4 Watchdog configuration register (WDTCFG0/1/2/3) 2) Corrected section 24.1.2 Reset control flag register (RESF) |
| | February 2024 | Correct the content of Chapter 6 and Chapter 22 |
| V1.0.0 | Mar 2024 | 1) Modify the section 12.5.5 register instructions 2) Modify the section 12.5.6 register instructions 3) Modify the section 12.5.7 register instructions 4) Modify the section 12.5.8 register instructions |
| | Sep 2024 | Revised the cover page |
| | Dec 2024 | 1) Modified the 18.3 PGA Structure diagram 2) Modified the errors in sections 2.2 and 6.2.17 |
| V1.0.1 | Apr 2025 | Modified the description in section 14.5.3 |