FEUL62Q2045-02

ML62Q2033/2035/2043/2045
User's Manual

Issue Date: September. 5, 2024

Notice

Precaution on using ROHM Products

1) When using ROHM Products, refer to the latest product information and ensure that usage conditions (absolute maximum ratings[*1], recommended operating conditions, etc.) are within the ranges specified. ROHM disclaims any and all liability for any malfunctions, failure or accident arising out of or in connection with the use of ROHM Products outside of such usage conditions specified ranges, or without observing precautions. Even if it is used within such usage conditions specified ranges, semiconductors can break down and malfunction due to various factors. Therefore, in order to prevent personal injury, fire or the other damage from break down or malfunction of ROHM Products, please take safety at your own risk measures such as complying with the derating characteristics, implementing redundant and fire prevention designs, and utilizing backups and fail-safe procedures.
   *1: Absolute maximum ratings: a limit value that must not be exceeded even momentarily.

2) The Products specified in this document are not designed to be radiation tolerant.

3) Descriptions of circuits, software and other related information in this document are provided only to illustrate the standard operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. And the peripheral conditions must be taken into account when designing circuits for mass production. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, and other related information.

4) No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of ROHM or any third party with respect to ROHM Products or the information contained in this document (including but not limited to, the Product data, drawings, charts, programs, algorithms, and application examples, etc.). Therefore, ROHM shall have no responsibility whatsoever for any dispute, concerning such rights owned by third parties, arising out of the use of such technical information.

5) ROHM intends our Products to be used in a way indicated in this document. Please be sure to contact a ROHM sales office if you consider the use of our Products in different way from original use indicated in this document. For use of our Products in medical systems, please be sure to contact a ROHM representative and must obtain written agreement. Do not use our Products in applications which may directly cause injuries to human life, and which require extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters, etc. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising by using the Product for purposes not intended by us without our prior written consent.

6) Please use the Products in accordance with any applicable environmental laws and regulations, such as the RoHS Directive. ROHM shall have no responsibility for any damages or losses resulting non-compliance with any applicable laws or regulations.

7) When providing our Products and technologies contained in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, including without limitation the US Export Administration Regulations and the Foreign Exchange and Foreign Trade Act.

8) Please contact a ROHM sales office if you have any questions regarding the information contained in this document or ROHM's Products.

9) This document, in part or in whole, may not be reprinted or reproduced without prior consent of ROHM.

(Note) “ROHM” as used in this document means ROHM Co., Ltd.

Other Precaution

1) All information contained in this document is subject to change for the purpose of improvement, etc. without any prior notice. Before purchasing or using ROHM Products, please confirm the latest information with a ROHM sales office.

2) ROHM has used reasonable care to ensure the accuracy of the information contained in this document, however, ROHM shall have no responsibility for any damages, expenses or losses arising from inaccuracy or errors of such information.

LTSZ08023・01・003

Notes for product usage

Notes on this page are applicable to the all ROHM microcontroller products.
For individual notes on each RHOM microcontroller product, refer to [Note] in the chapters of each user's manual.
The individual notes of each user's manual take priority over those contents in this page if they are different.

1. HANDLING OF UNUSED INPUT PINS
   Fix the unused input pins to the power pin or GND to prevent to cause the device performing wrong operation or increasing the current consumption due to noise, etc. If the handlings for the unused pins are described in the chapters, follow the instruction.

2. STATE AT POWER ON
   At the power on, the data in the internal registers and output of the ports are undefined until the power supply voltage reaches to the recommended operating condition and "L" level is input to the reset pin.
   On ROHM microcontroller products that have the power on reset function, the data in the internal registers and output of the ports are undefined until the power on reset is generated.
   Be careful to design the application system does not work incorrectly due to the undefined data of internal registers and output of the ports.

3. ACCESS TO UNUSED MEMORY
   If reading from unused address area or writing to unused address area of the memory, the operations are not guaranteed.

4. CHARACTERISTICS DIFFERENCE BETWEEN THE PRODUCT
   Electrical characteristics, noise tolerance, noise radiation amount, and the other characteristics are different from each microcontroller product.
   When replacing from other product to ROHM microcontroller products, please evaluate enough the apparatus/system which implemented ROHM microcontroller products.

5. USE ENVIRONMENT
   When using ROHM microcontroller products in a high humidity environment and an environment where dew condensation, take moisture-proof measures.

Notation

Classification	Notation	Description
Numeric value	XXh, XXH, 0xXX	Indicates a hexadecimal number.
Unit	word, W	1 word = 16 bits
	byte, B	1 byte = 8 bits
	nibble, N	1 nibble = 4 bits
	mega-, M	10^6
	kilo-, K	2^{10} = 1024
	kilo-, k	10^3 = 1000
	milli-, m	10^{-3}
	micro-, μ	10^{-6}
	nano-, n	10^{-9}
	second, s (lower case)	second
Terminology	"H" level	Indicates high level voltage V_{IH} and V_{OH} as specified by the electrical characteristics in the data-sheet.
	"L" level	Indicates low level voltage V_{IL} and V_{OL} as specified by the electrical characteristics in the data-sheet.
	SFR	Special function register. It is control register for system or peripherals.

- Register description

"R/W" indicates that Read/Write attribute. "R" indicates that data can be read and "W" indicates that data can be written. "R/W" indicates that data can be read or written.

MSB: The highest bit of 16-bit register
LSB: The lowest bit of 16-bit register

Registers that have a word symbol allow the word-access. If writing or reading the registers not using the word symbol, specify the even number addresses.

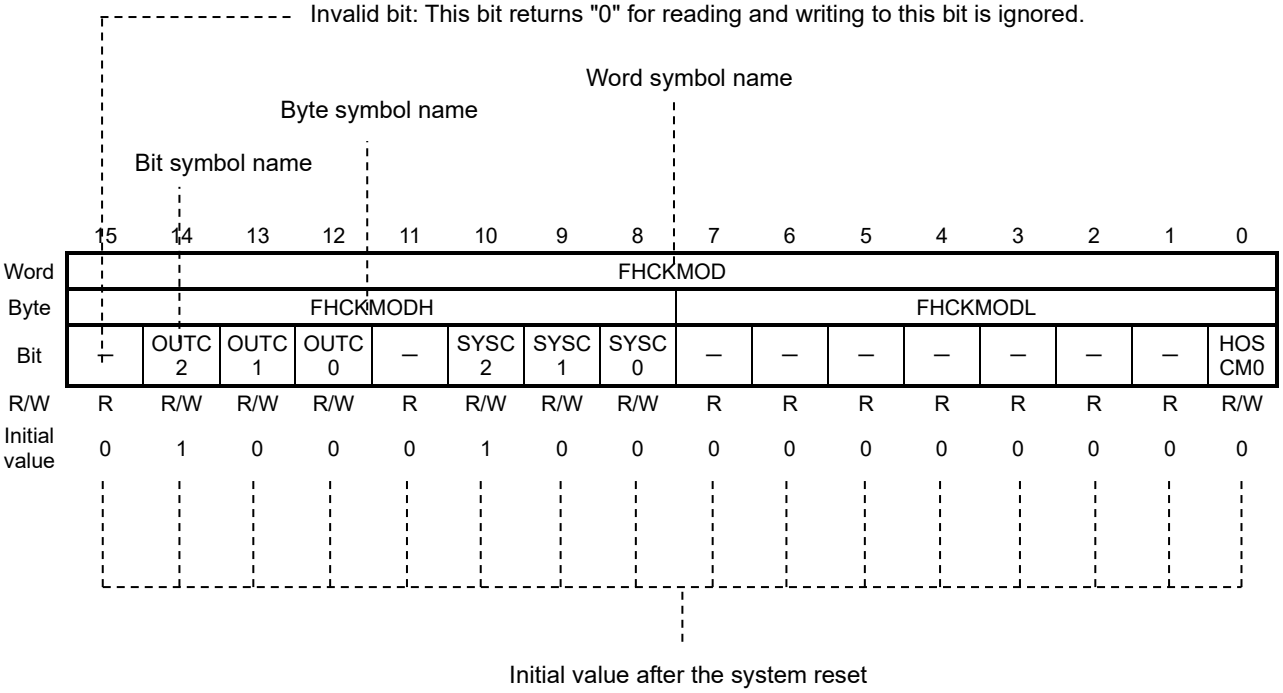


Table of Contents

Chapter 1 Overview

1. Overview

ML62Q2033/2035/2043/2045 are high performance CMOS 16-bit microcontrollers equipped with an 16-bit CPU nX-U16/100 and integrated with program memory (Flash memory), data memory (RAM), data Flash (Erase unit:128byte, Write unit:1byte) and rich peripheral functions such as the multiplier/divider, Clock generator, Operational timer (PWM, capture), Timer, General Purpose Ports, UART, I2C bus interface unit(Master, Slave), Successive approximation type 12bit A/D converter, 8bit D/A converter, PGA (Programable Gain Amp) and so on.
The CPU nX-U16/100 is capable of efficient instruction execution in 1-instruction 1-clock mode by pipeline architecture parallel processing.
The built-in on-chip debug function enables debugging and programming the software. Also, ISP (In-System Programming) function supports the Flash programming in production line.

- Applications

  Consumer and Industrial equipment (e.g., Household appliances, Housing equipment, Office equipment, Measurement instrumentation, etc.)

  Note:
  This product cannot be applicable for automotive use, automatic train control systems, and railway safety systems. Please contact ROHM sales office in advance if contemplating the integration of this product into applications that requires high reliability, such as transportation equipment for ships and railways, communication equipment for trunk lines, traffic signal equipment, power transmission systems, core systems for financial terminals and various safety control devices.

- Product list

  Table1-1 shows the combination of ML62Q2033/2035/2043/2045 memory variations and package type.

Table 1-1 Product List

Program memory	Data memory (RAM)	Data Flash	20pin TSSOP20	24pin WQFN24
32Kbyte	2Kbyte	4Kbyte	ML62Q2035	ML62Q2045
16Kbyte			ML62Q2033	ML62Q2043

1.1 Features

- CPU
  - 16-bit RISC CPU : nX-U16/100 (A35 core)
  - Instruction system : 16-bit length instructions
  - Instruction set : Transfer, arithmetic operations, comparison, logic operations,
    - multiplication/division, bit manipulations, bit logic operations, jump, conditional jump, call return stack, manipulations, arithmetic shift, and so on
  - Built-in On-chip debug function (connect to the ROHM on-chip debug emulator)
  - Minimum instruction execution time: 1 count of system clock
    Approximately 30.5μs/62.5ns (at 32.768kHz/16MHz system clock)

- Coprocessor for multiplication and division
  - Signed or Unsigned is selectable

Parameter	Expression	Operation time [cycle]
Multiplication	16bit × 16bit	4
Division	32bit ÷ 16bit	8
	32bit ÷ 32bit	16
Multiply-accumulate (non-saturating, non-saturating)	16bit × 16bit + 32bit	4

- Operating voltage and temperature
  - Operating voltage : V_{DD} = 4.5 to 5.5V
  - Operating temperature: -40°C to +105°C

- Flash memory

Parameter	Program memory area	Data Flash memory area
Erase/Write count	100cycles	10,000cycles
Write unit	16bit (2byte)	8bit (1byte)
Erase unit	16Kbyte/1Kbyte	all area/128byte
Erase/Write temperature (Ta)	0°C to +40°C	-40°C to +85°C

  - Background Operation (CPU can work while erasing and rewriting to the Data Flash memory area.)
  - The built-in on-chip debug function and ISP (In-System Programming) function enable Flash programming

    This product uses Super Flash® technology licensed from Silicon Storage Technology, Inc.
    Super Flash® is a registered trademark of Silicon Storage Technology, Inc.

- Data RAM area
  - Rewrite unit: 8bit/16bit (1byte/2byte)
  - Parity check function is available (interrupt or reset is generatable at Parity error)

- Clock generation circuit
  - Low-speed clock (LSCLK0)
  - Internal low-speed RC oscillation (RC32K) : Approx. 32.768kHz
  - High-speed source clock (HSOCLK) : Available for PWM generation circuit clock
  - PLL oscillation : 64MHz
  - High-speed system clock (HSCLK) : 16MHz, generated by dividing HSOCLK

- Reset
  - System Resets by reset input pin, Power-On Reset, Low Level Detector (LLD) reset, Watchdog timer (WDT) overflow, WDT invalid clear, RAM parity error reset, and Program Counter error reset (instruction access to unused ROM area)
  - Software reset by BRK instruction (reset CPU only)
  - Reset the peripherals individually/collectively by software
  -
- Power management
  - Two stand-by mode.
    - STOP mode (All clocks are stopped)
    - HALT mode (clocks for System are stopped)
  - Individual clock input control to the peripheral blocks by software
  - Clock gear: High-speed system clock frequency is changeable dynamically
  - (1/1, 1/2, 1/4, 1/8, 1/16, 1/32 of HSCLK)

- Interrupt controller
  - Non-maskable interrupt source : 1 (Internal sources: WDT)
  - Maskable interrupt sources : 22 (included the external interrupt 4 sources)
  - Four step interrupt levels
  - External interrupt ports（EXI） : 4 (selectable from Max.8 pins) with sampling filter
    - and edge (rise, fall, both) selection.

- General-purpose ports (GPIO)
  - I/O port : Max. 20 (Including pins for shared functions)
  - Carrier frequency output function (for IR communication)

- Watchdog timer (WDT): 1 channel
  - Overflow period : 8selectable (7.8, 15.6, 31.3, 62.5, 125, 500, 2000, 8000ms)
  - Selectable window function (enable or disable): configurable clear enable period (50% or 75% of overflow period) with invalid clear. When disable, interrupts the first overflow and resets the second overflow. When enable, reset occurs for the first overflow.
  - Selectable WDT operation : select Enable or Disable by code option
  - Selectable operation during HALT mode (Continue counting/Stop counting)
  - WDT counter operation monitoring function (Readable WDT counter)

- Operational timer: 6 channels
  - Various modes (Continuous, One shot, capture, and PWM mode)
  - Event trigger (external terminal, 16bit timer, operational timer, comparator) input is available
  - Selectable counter clock from various sources (LSCLK0, HSCLK (16MHz), HSOCLK (64MHz), divided by 1 to 8 of external pin input)
  - Logic AND output with several channel output (Operational timer output, comparator output, and external pin input) is available

- 16-bit General timers: 1 channel
  - Selectable counter clock from various sources (LSCLK, HSCLK (16MHz), and external clock divided by 1 to 8)

- UART (Half-duplex/Full-duplex communication mode): 2 channels
- Selectable from 5 to 8bit length, parity or no parity, odd parity or even parity, 1 stop bit or 2 stop bits, Positive logic or Negative logic, LSB first or MSB first
  - Sampling filter for receiving data and start bit
  - Built-in baud rate generator (HSCLK@16MHz: 300bps to 2Mbps, LSCLK: up to 2400bps)

- I^2C bus: 1 channel
  - Select from Master mode or Slave mode: 1channel. Master mode only: 1channel
  - Standard mode (100kbps), fast mode (400kbps) and 1Mbps mode (1Mbps)
  - 7bit address format
  - Master mode: Handshake (Clock synchronization), 10bit Master address format is supported
  - Slave mode: Clock stretch function,

- Successive approximation type 12bit A/D converter (SA-ADC): input 5 channels
  - Conversion time: Min. 1.375μs / ch (When the conversion clock frequency is 16MHz)
  - Reference voltages (V_{ADCREF}) are selectable from VDD pin input voltage or Internal reference voltage (V_{ADCREF}, Approx. 2.5V)
  - Dedicated result register for each channel
  - Interrupt by Continuous conversion and Trigger start

- Programable-Gain-Amp (PGA): 1 channel
  - Amplification factor : 4/ 8/16/ 32
  - Voltage input pin is selectable (AIN0/ AIN1/ AIN2/ AIN3)

- Analog comparator (CMP): 3 channels
  - Selectable interrupt from the comparator output (rising edge or falling edge) and sampling
  - Selectable reference voltage ($V_{CMPREFn\ n=0\ to\ 2}$) from external pin input, internal reference voltage (0.8V) ($V_{CMPREFI}$), and D/A converter

- D/A converter (DAC): 2 channels
  - Resolution : 8 bit
  - Output impedance :10kΩ (Typ.)
  - R-2R ladder type
  - Reference voltages (V_{DACREF}) are selectable from VDD pin input voltage or Internal reference voltage (0.8V) (V_{DACREF})
  -
- Low Level Detector (LLD): 1 channel
  - Reset generating
  - Sampling filter
  - Low power operation

- Safety Function
  - RAM/SFR guard
  - RAM parity error detection
  - ROM unused area access reset (instruction access)
  - WDT counter monitoring
  - SA-ADC test
  - Communication loop back test (UART, I^2C bus(master))
  - GPIO test

- Shipping package

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
20 pin plastic TSSOP	6.5 × 4.4 (6.5 × 6.4)	0.65	ML62Q2033-xxxTDZWARZ ML62Q2035-xxxTDZWARZ	ML62Q2033-xxxTDZWATZ ML62Q2035-xxxTDZWATZ
24 pin plastic WQFN	4.0 × 4.0 (-)	0.50	ML62Q2043-xxxGDZW5AY ML62Q2045-xxxGDZW5AY	ML62Q2043-xxxGDZW5BY ML62Q2045-xxxGDZW5BY

xxx: ROM code number, (NNN: ROM code is blank)

1.1.1 How To Read The Part Number

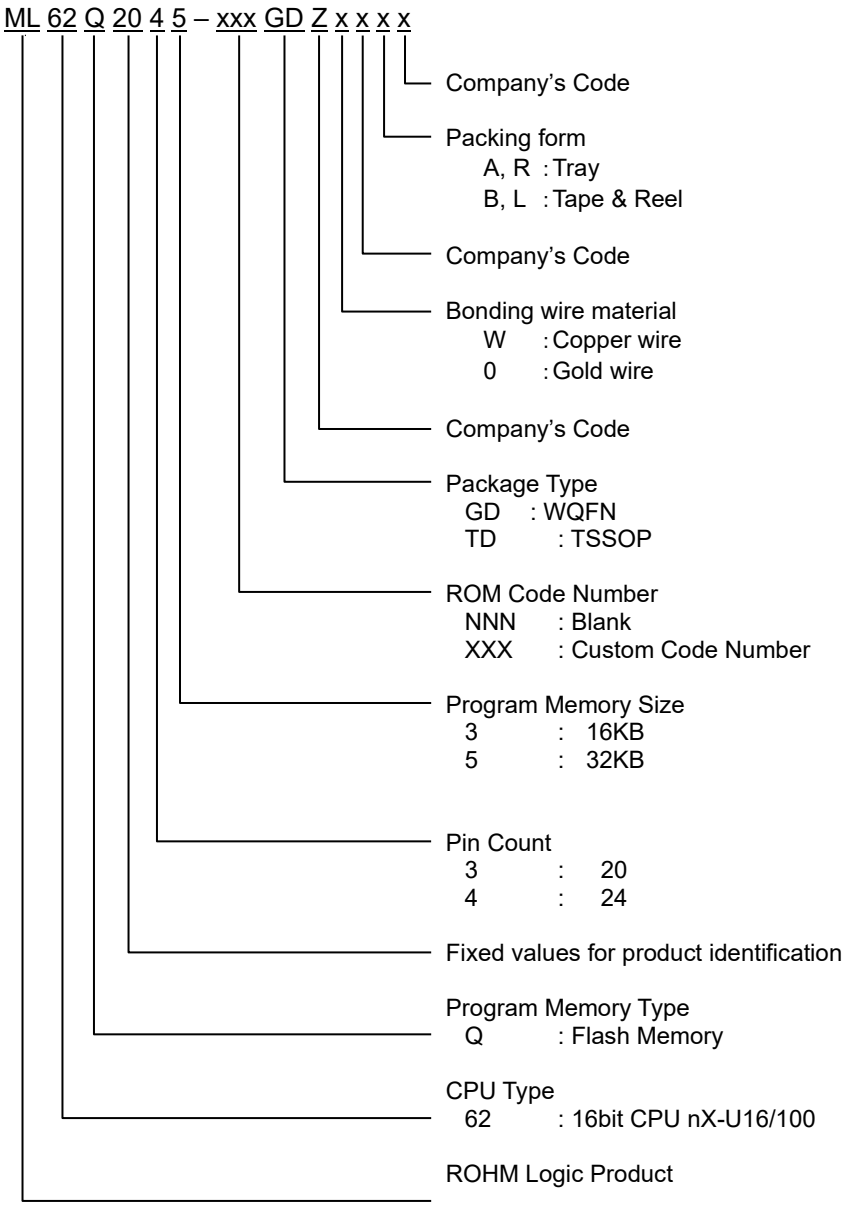


Figure 1-1 Part Number

1.1.2 Main Function List

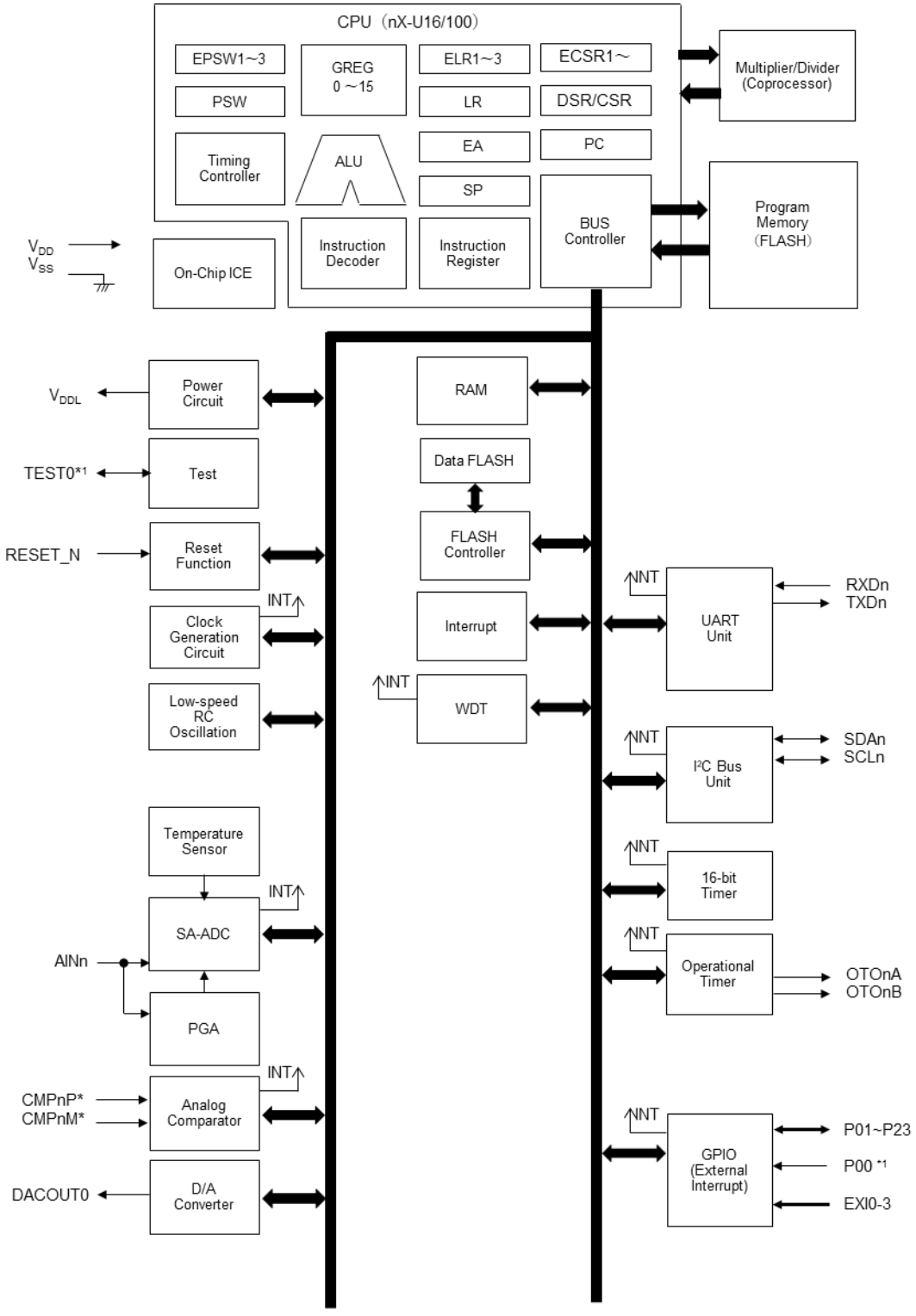
Table 1-2 Main Function List

Part number	Pin						Interrupt			Timer				Communication		Analog			
	Total pin	Power pin	Reset Input pin	General purpose I/O pin*1	General purpose I/O pin (LED drive is supported)	External interrupt pin	External interrupt source	Non maskable interrupt source	Internal maskable interrupt source	16bit Timer [ch]	16bit operational Timer [ch]	16bit operational Timer [Port]	Watchdog Timer [ch]	UART [ch]	I²C bus unit (Master / Slave) [ch]	12bit Successive type A/D converter [ch]	Analog Comparator [ch]	D/A converter [dh]	PGA [ch]
ML62Q2033	20	3	1	1	15	8	4	1	18	1	6	10	1	2	1	5	3	2	1
ML62Q2035																			
ML62Q2043	24				19							13							
ML62Q2045																			

*1: Shared with pins debug input.

1.2 Block Diagram

1.2.1 Block Diagram of ML62Q2033/2035/2043/2045



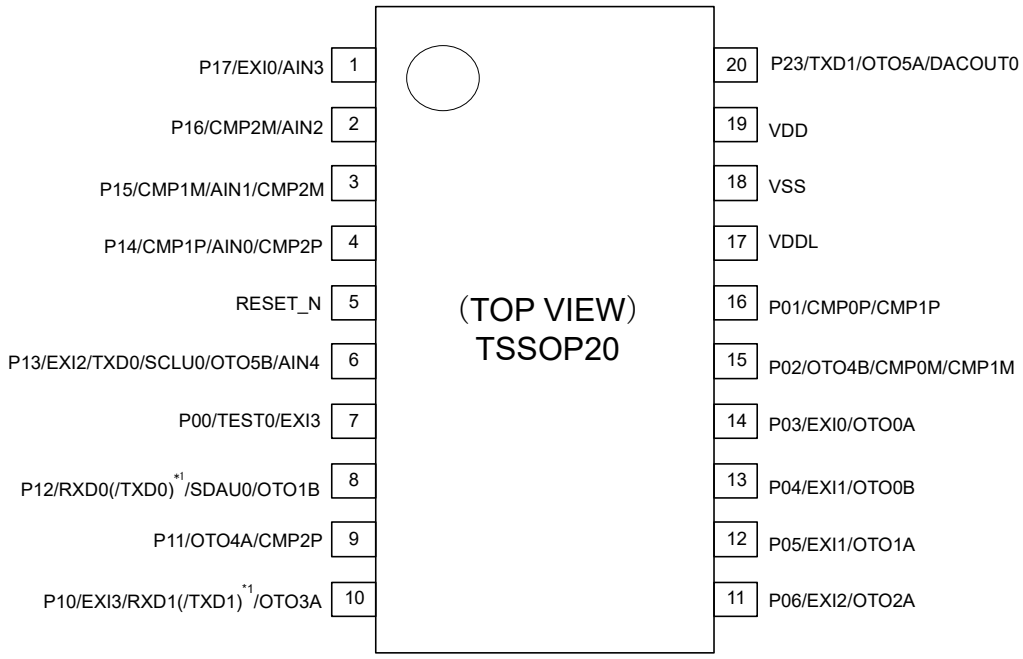
*1 : Not available as the input port when connecting to the on-chip debug emulator.

Figure 1-2 Block Diagram

1.3 Pin

1.3.1 Pin Layout

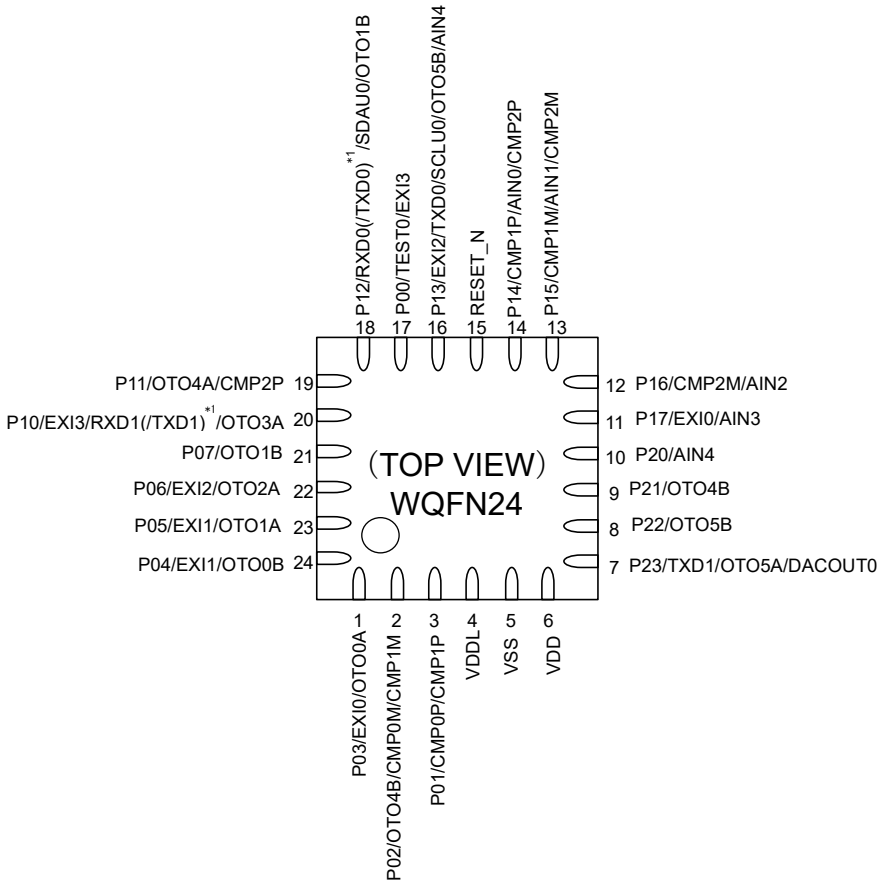
1.3.1.1 ML62Q2035/2033 : 20 pin TSSOP



*1 : RXDn pin is shareable with TXDn pin by SFR setting

Figure 1-3-1 20 pin TSSOP

1.3.1.2 ML62Q2045/2043 : 24 pin WQFN



DIE PAD = NC

*1 : RXDn pin is shareable with TXDn pin by SFR setting

Figure 1-3-2 24 pin WQFN

1.3.2 Pin List

Table 1-3 shows the pin list of This LSI.

Table 1-3 Pin List

Pin No.		Pin name	1st function	2nd function	3rd function	4th function	5th function	6th function	7th function	8th function
ML62Q203x	ML62Q204x		GPI/ EXI	UART*	I²C	OTM	CMP/DAC	ADC	CMP	CMP/ADC
19	6	VDD	-	-	-	-	-	-	-	-
18	5	VSS	-	-	-	-	-	-	-	-
17	4	VDDL	-	-	-	-	-	-	-	-
16	3	P01	-	-	-	-	CMP0P	-	CMP0P/ CMP1P	CMP0P
15	2	P02	-	-	-	OTO4B	CMP0M	-	CMP0M/ CMP1M	CMP0M
14	1	P03	EXI0	-	-	OTO0A	-	-	-	-
13	24	P04	EXI1	-	-	OTO0B	-	-	-	-
12	23	P05	EXI1	-	-	OTO1A	-	-	-	-
11	22	P06	EXI2	-	-	OTO2A	-	-	-	-
-	21	P07	-	-	-	OTO1B	-	-	-	-
10	20	P10	EXI3	RXD1*1 (/TXD1)*2	-	OTO3A	-	-	-	-
9	19	P11	-	-	-	OTO4A	CMP2P	-	CMP2P	CMP2P
8	18	P12	-	RXD0*1 (/TXD0)*2	SDAU0	OTO1B	-	-	-	-
7	17	P00/TEST0	EXI3	-	-	-	-	-	-	-
6	16	P13	EXI2	TXD0*1	SCLU0	OTO5B	-	AIN4	-	-
5	15	RESET_N	-	-	-	-	-	-	-	-
4	14	P14	-	-	-	-	CMP1P	AIN0	CMP1P/ CMP2P	AIN0 /CMP1P
3	13	P15	-	-	-	-	CMP1M	AIN1	CMP1M/ CMP2M	AIN1 /CMP1M
2	12	P16	-	-	-	-	CMP2M	AIN2	CMP2M	AIN2 /CMP2M
1	11	P17	EXI0	-	-	-	-	AIN3	-	-
-	10	P20	-	-	-	-	-	AIN4	-	-
-	9	P21	-	-	-	OTO4B	-	-	-	-
-	8	P22	-	-	-	OTO5B	-	-	-	-
20	7	P23	-	TXD1*1	-	OTO5A	DACOUT0	-	-	-
-	DIE	NC	-	-	-	-	-	-	-	-

*1: The UARTpin use with a combination of the same suffix pins
*2: RXDn pin is shareable with TXDn pin by SFR setting

1.3.3 Pin Description

Table 1-4 shows the pin list categorized by the function.
"I/O" Field in the below table define the pin type ("-" : power supply pin, "I" : Input pin, "O" : Out put pin, "I/O" bi-directional pin)

Table 1-4 Pin Description

Function	Functional pin name	LSI pin name	I/O	Description
Power	-	VSS	-	Negative power supply pin (-) Define this terminal potential as V_{SS}.
	-	VDD	-	Positive power supply pin (+). Connect a capacitor C_V (more than 1µF) between this pin and VSS. Define this terminal potential as V_{DD}
	-	VDDL	-	Power supply for internal logic (internal regulator's output). Connect a capacitor C_L (1µF) between this pin and VSS.
Debug ISP	TEST0	P00/ TEST0	I/O	Input/output for testing This pin which is shared with P00 is used as on-chip debug interface and ISP function and is initialized as pull-up input mode by the system reset.
Reset	RESET_N	RESET_N	I	Reset input. Appling this pin "L" level shifts MCU to system reset mode. Appling this pin "H" level shifts MCU to program running mode. No pull-up resistor is built-in.
General input port (GPI)	P00	P00/ TEST0	I/O	General purpose input. - Input with Pull-up (initial value) - Input without Pull-up Not available as general inputs when using the on-chip debug interface or ISP function.
General port (GPIO)	P01 to P07	P01 to P07	I/O	General purpose input/output - High-impedance (initial value) - Input with Pull-up - Input without Pull-up - CMOS output - N channel (N-ch) open drain output - P channel (P-ch) open drain output
	P10 to P17	P10 to P17		
	P20 to P23	P20 to P23		
Career frequency output	-	P13 P23	O	Career frequency output
External Interrupt (1st function)	EXI0	P03 P17	I	External Maskable Interrupt 0 input
	EXI1	P04 P05		External Maskable Interrupt 1 input
	EXI2	P06 P13		External Maskable Interrupt 2 input
	EXI3	P00 P10		External Maskable Interrupt 3 input
Operational Timer (4th function)	OTO0A	P03	O	Operational Timer 0 A output
	OTO0B	P04		Operational Timer 0 B output
	OTO1A	P05		Operational Timer 1 A output
	OTO1B	P07 P12		Operational Timer 1 B output
	OTO2A	P06		Operational Timer 2 A output
	OTO3A	P10		Operational Timer 3 A output
	OTO4A	P11		Operational Timer 4 A output
	OTO4B	P02 P21		Operational Timer 4 B output
	OTO5A	P23		Operational Timer 5 A output
	OTO5B	P13 P22		Operational Timer 5 B output
I^2C Bus (3rd function)	SCLU0	P13	I/O	I^2C Unit0 Clock input/output
	SDAU0	P12		I^2C Unit0 Data input/output

Function	Functional pin name	LSI pin name	I/O	Description
UART (2nd function)	RXD0	P12	I	UART0 received data input
	TXD0	P13	O	UART0 transmission data output
	RXD1	P10	I	UART1 received data input
	TXD1	P23	O	UART1 transmission data output
Successive approximation type A/D converter (SA-ADC) (6/8th function)	AIN0 to AIN4	P14 P15 P16 P17 P13 P20	I	SA-ADC channel 0 to 4 analog input
D/A converter (5th function)	DACOUT0	P23	O	D/A converter 0 output (select by SFR)
Analog comparator (5/7/8th function)	CMP0P	P01	I	Analog comparator 0 Noninverting input
	CMP0M	P02		Analog comparator 0 Inverting input
	CMP1P	P01 P14		Analog comparator 1 Noninverting input
	CMP1M	P02 P15		Analog comparator 1 Inverting input
	CMP2P	P11 P14		Analog comparator 2 Noninverting input
	CMP2M	P15 P16		Analog comparator 2 Inverting input

1.3.4 Termination of Unused Pins

Table 1-5 shows the processing of unused pins.

Table 1-5 Termination of unused pins

Pin	pin termination
RESET_N	Connect to VDD
P00/TEST0	Open with the initial condition of pulled-up input mode
P01 to P07 P10 to P17 P20 to P23	Open the pins with the initial condition of Hi-impedance (input/output invalid) mode.

[Note]

- **Terminate unused input pins according to the Table 1-5 in order to avoid unexpected through-current in the pins.**

Chapter 2 CPU and Memory Space

2. CPU and Memory Space

2.1 General Description

This LSI has 16-bit CPU nX-U16/100 (A35 core), the multiplier/divider in the coprocessor, flash memory in the program memory space, and RAM and data flash in the data memory space.
In addition, it has the built-in remap function that remaps a 4 Kbyte area in the program memory space.
Table 2-1 to 2-3 show the memory size of the program memory space and the data memory space as well as the CPU memory model. For details of memory model, see "nX-U16/100 Core Instruction Manual".

Table 2-1 Program Memory Space and Data Memory Space

Product name	Program memory space	Data memory space	Data flash size	Memory model
	ROM size	RAM size		
ML62Q2033 ML62Q2043	16 Kbyte	2 Kbyte	4 Kbyte	SMALL
ML62Q2035 ML62Q2045	32 Kbyte			

2.2 CPU nX-U16/100

nX-U16/100 has following features. See "nX-U16/100 Core Instruction Manual" for details.

- Various instruction sets
  - Instructions for data transfers, arithmetic, comparison, logic operations, multiplication/division, bit manipulation, bitwise logic operations, branches, conditional branches, call/return stack manipulation, and arithmetic shifts
- Variety of addressing modes
  - Register addressing
  - Register indirect addressing
  - Stack pointer addressing
  - Control register addressing
  - EA register indirect addressing
  - General-purpose register indirect addressing
  - Direct addressing
  - Register indirect bit addressing
  - Direct bit addressing
- Memory space
  - Program memory space
  - Data memory space
- Interrupts
  - Dedicated emulator interrupt
  - Non-maskable interrupt
  - Maskable interrupt
  - Software interrupt

2.2.1 Wait Mode and No-wait Mode

nX-U16/100 has two CPU operation modes.
This LSI only supports no-wait mode.

Table 2-2 Maximum Operating Frequency

PLL reference frequency	Maximum operating frequency of peripheral circuit	Maximum operating frequency of CPU
		No-wait mode
64MHz	16MHz*1 *1: Only the operational timer can operate at up to 64 MHz	16MHz

- Wait mode (This LSI is not supported)
  In this mode, instruction codes read from the program memory are stored into the built-in buffer. The CPU can work at high speed to read the instructions from the buffer. In contiguous address instruction processing, the instructions can be executed without a wait time for storing them in the buffer. In branch instruction processing, the number of execution cycles increases due to a wait time for storing the instructions in the buffer.
- No-wait mode
  This mode allows the CPU to directly execute instruction codes read from the program memory without involving the buffer. This mode minimizes the number of instruction execution cycles.

See Appendix C "Instruction execution cycle" for the number of instruction execution cycles in wait and no-wait modes.

2.2.2 Notes When Executing SB/RB Instruction

The bit access SB/RB instruction reads in bytes from a register containing the target bits, generates the byte data while rewriting only the values of the target bits, then writes it in bytes.
If an SB/RB instruction is executed to a register where multiple bits are placed, bits not targeted for the SB/RB instruction are rewritten with the values read at that time.
Note that the SB/RB instruction may rewrite the state of bits not targeted for the SB/RB instruction if it is executed to a register where values of some bits change depending on the hardware state.

2.2.3 Notes on the Description of Read-modify-write

When reading values from SFR and changing only some of the values and writing them back (read-modify-write), C compiler may convert it to a bit-access instruction. (Even if the change is two bits, it may be converted to two bit-access instructions.) Therefore, there are cases where you think you are writing at the same time, but you are not, and cases where you think you are doing word-access, but it is converted to bit-access.
If you do not want to be converted to a bit-access instruction, you can avoid it by the following description.

Example of a description that is converted to a bit-access instruction:

```
SFR &= 0xFFFE;

SFR |= 0x0081;
```

Converted to RB SFR.0 ;

Converted to SB SFR.7 ;
SB SFR.0 ;

Example of a description that is not converted to a bit-access instruction:

```
volatile unsigned short vald;
vald = SFR;
SFR = vald & 0xFFFE;

vald = SFR;
SFR = vald | 0x0081 ;
```

The conversion to bit-access instructions can be avoided by assigning the variable once to a volatile-qualified variable.

2.3 Coprocessor

This LSI has the built-in multiplier/divider in the coprocessor.
The multiplier/divider is operated using coprocessor data transfer instructions of the CPU. For coprocessor data transfer instructions, see "nX-U16/100 Core Instruction Manual".

2.3.1 Multiplier/Divider

The multiplier/divider has following arithmetic functions:

- Multiplication : 16 bit × 16 bit (operation time 4 cycles)
- Division : 32 bit ÷ 16 bit (operation time 8 cycles)
- Division : 32 bit ÷ 32 bit (operation time 16 cycles)
- Multiply-accumulate (non-saturating) : 16 bit × 16 bit + 32 bit (operation time 4 cycles)
- Multiply-accumulate (saturating) : 16 bit × 16 bit + 32 bit (operation time 4 cycles)
- Signed or unsigned operation setting
- In a saturating multiply-accumulate operation, the result is fixed to 0x7FFF_FFFF for a positive number and 0x8000_0000 for a negative number when it is out of the expressible range.

See user's manual for the multiplication/division library using the multiplier/divider.

2.3.2 List of Coprocessor General-purpose Registers

The coprocessor general-purpose registers are byte type and readable or writable as word type registers (CERn), double word type registers (CXRn), or quad word type registers (CQRn) combining the consecutive registers.

Table 2-3 List of coprocessor general-purpose registers

Address	Coprocessor general-purpose register	Symbol name				R/W	Initial value
		Byte	Word	Double word	Quad word		
-	A register L	CR0	CER0	CXR0	CQR0	R/W	0x00
-	A register H	CR1				R/W	0x00
-	B register L	CR2	CER2			R/W	0x00
-	B register H	CR3				R/W	0x00
-	C register L	CR4	CER4	CXR4		R/W	0x00
-	C register H	CR5				R/W	0x00
-	D register L	CR6	CER6			R/W	0x00
-	D register H	CR7				R/W	0x00
-	Operation mode register	CR8	CER8	CXR8	CQR8	R/W	0x00
-	Operation status register	CR9				R/W	0x00
-	-	CR10	CER10			R/W	0x00
-	-	CR11				R/W	0x00
-	-	CR12	CER12	CXR12		R/W	0x00
-	-	CR13				R/W	0x00
-	-	CR14	CER14			R/W	0x00
-	Coprocessor ID register	CR15				R	0x81

CR0 to CR7 are registers to store the setting of the input values of operations and operation results.
CR8 is a register to set each operation mode (signed, unsigned) and to enable/disable the operation.
CR9 is a register to store the status of each operation result.
CR15 is a register to indicate coprocessor ID.
CR10 to CR14 have no function. Reading them returns "0x00". These registers are not writable.

2.3.2.1 A, B, C, D Registers (CR0 to CR7)

These registers store the input values of operations and operation results.
These are byte type registers and can be accessed as a word type register (CERn), double word type register (CXRn), or quad word type register (CQRn) combining the consecutive registers.
The bit symbols are unavailable to use in the software.

Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER0															
Byte	CR1								CR0							
Bit	areg15	areg14	areg13	areg12	areg11	areg10	areg9	areg8	areg7	areg6	areg5	areg4	areg3	areg2	areg1	areg0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER2															
Byte	CR3								CR2							
Bit	breg15	breg14	breg13	breg12	breg11	breg10	breg9	breg8	breg7	breg6	breg5	breg4	breg3	breg2	breg1	breg0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER4															
Byte	CR5								CR4							
Bit	creg15	creg14	creg13	creg12	creg11	creg10	creg9	creg8	creg7	creg6	creg5	creg4	creg3	creg2	creg1	creg0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER6															
Byte	CR7								CR6							
Bit	dreg15	dreg14	dreg13	dreg12	dreg11	dreg10	dreg9	dreg8	dreg7	dreg6	dreg5	dreg4	dreg3	dreg2	dreg1	dreg0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-4 shows assignment of an input and results.
As soon as the data is written in register CR7, operation is started.
In a saturating multiply-accumulate operation, the result is fixed to 0x7FFF_FFFF for a positive number and 0x8000_0000 for a negative number when it is out of the expressible range.
In a signed operation, each of the most significant bits of input and output is a sign.

Table 2-4 assignment of an input and results

Quad word symbol		CQR0							
Double word symbol		CXR4				CXR0			
Word symbol		CER6		CER4		CER2		CER0	
Byte symbol		CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Multiplication 16 bit x 16 bit	Input	Multiplicand [15:0]		Multiplier [15:0]		-			
	Result	-		-		Product [31:0]			
Division 32 bit ÷ 16 bit	Input	Divisor [15:0]		-		Dividend [31:0]			
	Result	-		Remainder [15:0]		Quotient [31:0]			
Division 32 bit ÷ 32 bit	Input	Divisor [31:0]				Dividend [31:0]			
	Result	Remainder [31:0]				Quotient [31:0]			
Multiply-accumulate (non-saturating) 16 bit x 16bit + 32 bit	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]			
	Result	-		-		Multiply-accumulate [31:0]			
Multiply-accumulate (saturating) 16 bit x 16bit + 32 bit	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]			
	Result	-		-		Multiply-accumulate [31:0]			

"-" indicates that the previous value is retained.

2.3.2.2 Operation Mode Register (CR8), Operation Status Register (CR9)

The operation mode register (CR8) is a coprocessor general-purpose register to set the operation mode and enables/disables the operation.
The operation status register (CR9) is a register to store the status of each operation result.
CR8 and CR9 are byte type registers and they can be accessed as a word type register (CERn), double word type register (CXRn), or quad word type register (CQRn) combining the consecutive registers.
The bit symbols are unavailable to use in the software.

Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER8															
Byte	CR9								CR8							
Bit	c	z	s	ov	q	-	-	use	clen	-	-	sign	-	clmod2	clmod1	clmod0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	c	This becomes "1" if the operation result is carried or the divisor is 0 in the division mode. The value is updated in each operation. In addition, a value can be written.
14	z	This becomes "1" if the operation result is "0". The value is updated in each operation. In addition, a value can be written.
13	s	This becomes "1" if the operation result is a negative number. For a multiply-accumulate (saturating/non-saturating) operation, this indicates the state of the most significant bit in the operation result. The value is updated in each operation. In addition, a value can be written.
12	ov	This becomes "1" if the operation result exceeds the range expressible by two's complement. The value is updated every time the operation is executed. In addition, a value can be written.
11	q	This becomes "1" for the saturated result of a saturating multiply-accumulate operation. The value is held in the next operation. To initialize it to "0", it is necessary to write "0".
8	use	A bit to indicate that the operation is in progress. 0: Operation under suspension (initial value) 1: Operating
7	clen	A bit to enable/disable the operation. If the clen bit is cleared to "0" during an operation, the next operation is disabled after completion of the current one. 0: Operation disabled (initial value) 1: Operation enabled
4	sign	A bit to set the sign operation. 0: Unsigned operation (initial value) 1: Signed operation
2 to 0	clmod2 to clmod0	Bits to select the operation mode. 000: Multiplication 16 bit × 16 bit (initial value) 001: Division 32 bit ÷ 16 bit 010: Multiply-accumulate (non-saturating) 16 bit × 16 bit + 32 bit 011: Multiply-accumulate (saturating) 16 bit × 16 bit + 32 bit 100: No operation function 101: Division 32 bit ÷ 32 bit 110: No operation function 111: No operation function

Table 2-5 shows values to be set to CR8 register for execution of each operation mode.
Table 2-6 shows flags changing during each operation.

Table 2-5 Configured CR8

Value set to CR8	Signed	Unsigned
Multiplication 16 bit ×16 bit (initial value)	0x90	0x80
Division 32 bit ÷ 16 bit	0x91	0x81
Division 32 bit ÷ 32 bit	0x95	0x85
Multiply-accumulate (non-saturating) 16 bit × 16 bit + 32 bit	0x92	0x82
Multiply-accumulate (saturating) 16 bit × 16 bit + 32 bit	0x93	0x83

Table 2-6 Flag of CR9

Operation mode	sign	c	z	s	ov	q
Multiplication 16 bit × 16 bit	1 (signed)	-	●	●	-	-
	0 (unsigned)	-	●	-	-	-
Division 32 bit ÷ 16 bit	1 (signed)	●	●	●	●	-
	0 (unsigned)	●	●	-	-	-
Division 32 bit ÷ 32 bit	1 (signed)	●	●	●	●	-
	0 (unsigned)	●	●	-	-	-
Multiply-accumulate (non-saturating) 16 bit ×16 bit + 32 bit	1 (signed)	●	●	●	●	-
	0 (unsigned)	●	●	●	●	-
Multiply-accumulate (saturating) 16 bit × 16 bit + 32 bit	1 (signed)	●	●	●	●	●
	0 (unsigned)	●	●	●	●	●

●: Varies depending on the result. -: Retains the previous value.

2.3.2.3 Coprocessor ID Register (CR15)

CR15 is a read-only register to indicate coprocessor ID.
The value in CR15 register is fixed to "0x81".
It is a byte type register and it can be accessed as a word type register (CERn), double word type register (CXRn), or quad word type register (CQRn) combining the consecutive registers.
The bit symbols are unavailable to use in the software.

Access: R
Access size: 8/16 bit
Initial value: 0x8100

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER14															
Byte	CR15								CR14							
Bit	copid7	copid6	copid5	copid4	copid3	copid2	copid1	copid0	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

2.3.3 How to Use Multiplier/Divider

Set the calculation mode in the calculation mode register (CR8) and write the input value to CR0~CR7. When you write to CR7, the operation starts. After the calculation time has elapsed, the result is read from CR0~CR7. See Table 2-7 for the calculation mode settings and calculation time.

Table 2-7 Calculation mode and calculation time

Operation mode	CR8[2:0]		CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0	operation time
Multiplication	000b	Input	Multiplicand [15:0]		Multiplier [15:0]		-				4 cycles
16bit×16bit		Result	-		-		Product [31:0]				
Division	001b	Input	Divisor [15:0]		-		Dividend [31:0]				8 cycles
32bit÷16bit		Result	-		Remainder [15:0]		Quotient [31:0]				
Division	101b	Input	Divisor [31:0]				Dividend [31:0]				16 cycles
32bit÷32bit		Result	Remainder [31:0]				Quotient [31:0]				
Multiply-accumulate (non-saturating)	010b	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]				4 cycles
16bit×16bit+32bit		Result	-		-		Multiply-accumulate [31:0]				
Multiply-accumulate (saturating)	011b	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]				4 cycles
16bit×16bit;32bit		Result	-		-		Multiply-accumulate [31:0]				

"-" indicates that the previous value is retained.

The following is an example of how to perform multiplication.

Example, multiplication of 0x1234H × 0x0AA55H

```
MOV R2 ,#55H ; Setting the Multiplier
MOV R3 ,#0AAH ;
MOV R0 ,#34H ; Setting the multiplicand
MOV R1 ,#12H ;
MOV CR4 ,R0 ; Transferring multiplier [7:0]
MOV R0 ,#90H ; Setting Multiply mode
MOV CR8 ,R0 ; Setting signed multiply mode
MOV CR5 ,R1 ; Transferring multiplier [15:8]
MOV CR6 ,R2 ; Transferring multiplicand [7:0]
MOV CR7 ,R3 ; Transferring multiplicand [15:8],    starting of calculation
NOP ; Calculation in progress. Waiting for the end of the calculation (1 clock)
NOP ; Calculation in progress. Waiting for the end of the calculation (1 clock)
NOP ; Calculation in progress. Waiting for the end of the calculation (1 clock)
NOP ; Calculation in progress. Waiting for the end of the calculation (1 clock) ; End of calculation
READ:
MOV R0 ,CR0 ; Transferring multiplication [7:0]
MOV R1 ,CR1 ; Transferring multiplication [15:8]
MOV R2 ,CR2 ; Transferring multiplication [23:16]
MOV R3 ,CR3 ; Transferring multiplication [31:24]
```

[Note]

- **Changing to STOP/HALT mode during calculation is prohibited. When changing to STOP/HALT mode after using the multiplier and divider, wait until the calculation time in Table 2-7 has elapsed.**

2.4 Memory Space

The memory space refers to the address range of the memory that can be specified from the CPU. Figure 2-1 shows the general scheme of the memory space. The memory space of the nX-U16/100 is composed of the program memory space and data memory space. The memory space is managed as one segment consists of 64 Kbyte.
The program memory space can be read with a memory access instruction through the ROM window area or the mirror area. To read the data memory space, a memory access instruction is used.
The ROM window is an area provided to read the program memory space segment 0 through a memory access instruction. In reading the program memory space from this area, it is expected to gain the advantage of data compression and improvement in access speed because it is not required to specify DSR of the data memory space. In addition, the mirror area is provided to read program memory space segments 0 to 7 through a memory access instruction. There is no address limitation when reading the program memory space from this area.

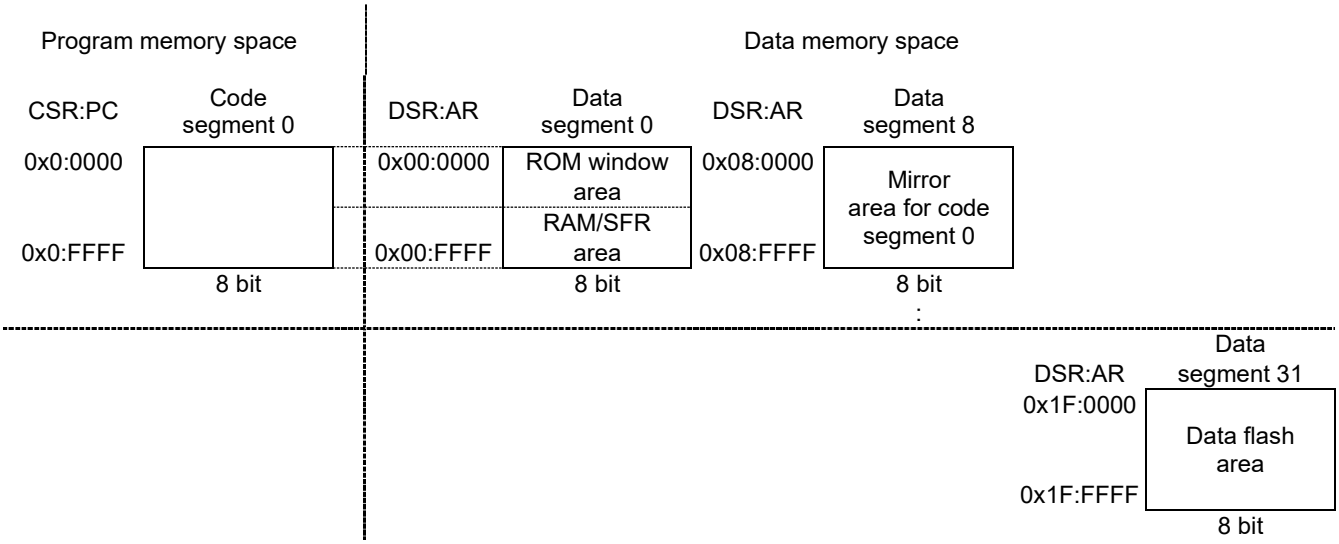


Figure 2-1 General Scheme of Memory Space

2.5 Program Memory Space

The program memory space is an area to store the program code, vector table, and Code Options.
The program memory space is specified by 20 bits (CSR:PC) consisting of higher 4 bits as code segment register (CSR) and lower 16 bits as program counter (PC).
The vector table area is used as the reset vector, hardware interrupt vector, and software interrupt vector. Unused software interrupt vector area is available as a program code area.
The Code Option area can be used to choose the CPU operation mode, PLL reference frequency, watchdog timer (WDT) operation mode, unused ROM area access reset enabled/disabled, and remapping function enabled/disabled.
The program code, vector table, and Code Option areas can be read from the ROM window area or the mirror area of the data memory space by executing the memory access instruction.
Figures 2-2 show the program memory space configuration of each product.

[Note]

- **The CSR of this LSI is fixed at "0".**
- **The Code Option area (48 bytes) is not available for the program code area. For details of Code Option settings, see Chapter 30 "Code Option" and make sure the setting data is correct.**
- **It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using the generation tool of the ROM code data. See its manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

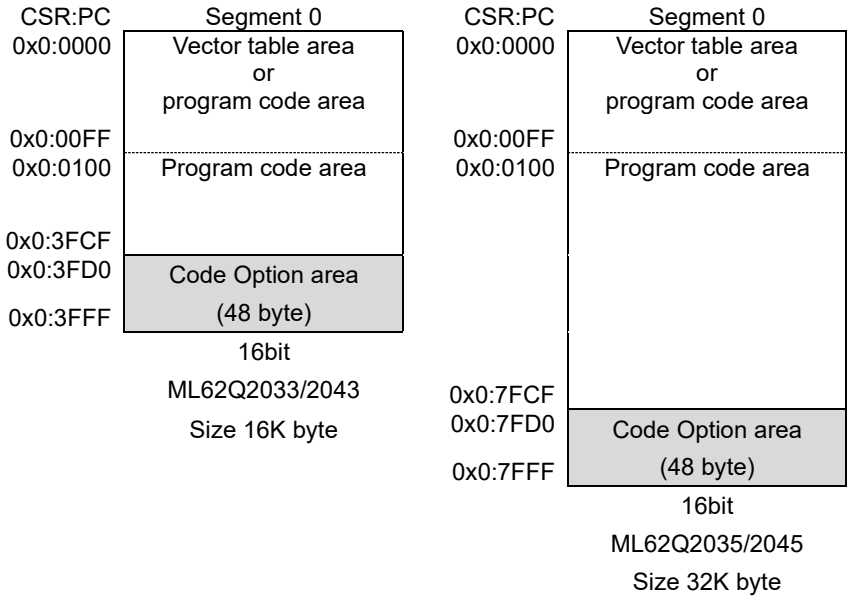


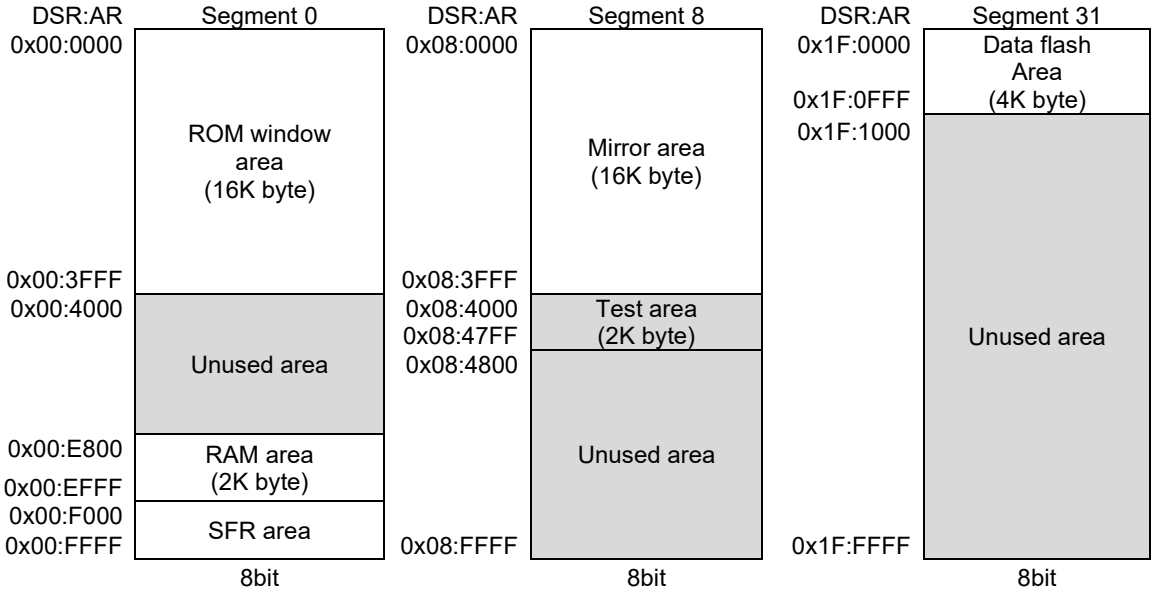
Figure 2-2 Configuration of Program Memory Space 1

2.6 Data Memory Space

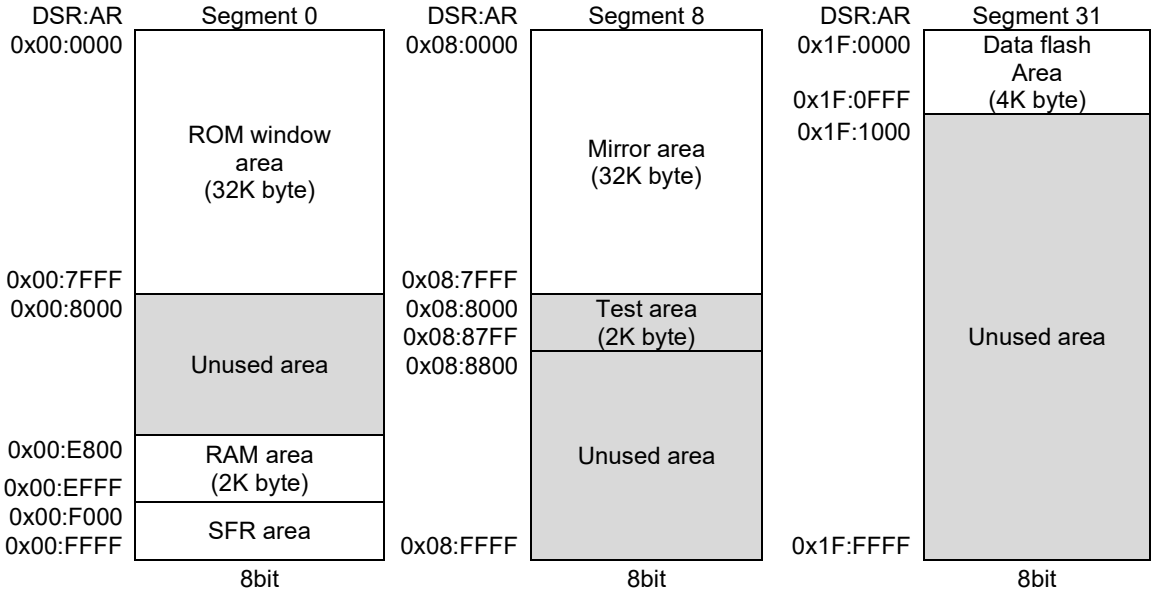
The data memory space consists of the segment 0 for ROM window area, RAM area, SFR area, segments 1 to 15 for mirror area, test area, and segment 31 for the data flash area.
The data memory stores 8-bit data and is specified by 21 bits consisting of higher 5 bits as the data segment register (DSR) and lower 16 bits as data address (address register: AR) specified by each instruction.
The segment 0 of program memory space and the segment of data memory space are in different space, but the segment 0 of program memory space is readable through the ROM window area of the data memory space.
The segment 1 to 7 are mirror area of segment 1 to 7 in the program memory space. The segment 8 to 15 are mirror area of segment 0 to 7 in the program memory space.
The 1K byte of test area includes device-specific data,
Figures 2-3 show the configuration of the data memory space of ML62Q2500 series products. Other segments not shown in the figures are unused areas.
The data flash area in segment 31 can only read bytes.

[Note]

- **The contents of the RAM area are undefined at power-on and system reset. Initialize this area by the software.**
- **Do not read/write unused areas to prevent the CPU works incorrectly.**



ML62Q2033/2043
Figure 2-3-1 Configuration of Data Memory Space1



ML62Q2035/2045
Figure 2-3-2 Configuration of Data Memory Space2

[Note]
- **Read the data flash area in bytes.**

2.7 Description of Registers

2.7.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF000	Data segment register	DSR	-	R/W	8	0x00
0xF0A0	Flash remap address register	REMAPADD	-	R/W	8	*1
0xF0A4	Reserved	-	-	R/W	8	0x00
0xF0A6	Reserved	-	-	R/W	8	0x00

*1: The initial value depends on Code Option settings. See Section 30.2.4 “Code Options 2 (CODEOP2)” for details of Code Option settings.

2.7.2 Data Segment Register (DSR)

DSR is a SFR to specify a data segment.
See "nX-U16/100 Core Instruction Manual" for details of DSR.

Address: 0xF000(DSR)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DSR							
Bit	-	-	-	-	-	-	-	-	-	-	-	DSR4	DSR3	DSR2	DSR1	DSR0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4 to 0	DSR4 to DSR0	00000: Data segment 0 (initial value) 00001: Mirror area of code segment 1 00010: Mirror area of code segment 2 00011: Mirror area of code segment 3 00100: Mirror area of code segment 4 00101: Mirror area of code segment 5 00110: Mirror area of code segment 6 00111: Mirror area of code segment 7 01000: Data segment 8 (mirror area of code segment 0) 01001: Data segment 9 (mirror area of code segment 1) 01010: Data segment 10 (mirror area of code segment 2) 01011: Data segment 11 (mirror area of code segment 3) 01100: Data segment 12 (mirror area of code segment 4) 01101: Data segment 13 (mirror area of code segment 5) 01110: Data segment 14 (mirror area of code segment 6) 01111: Data segment 15 (mirror area of code segment 7) :::::::::::: Data segments 16 to 30 (unused areas) 11111: Data segment 31 (data flash area)

2.7.3 Flash Remap Address Register (REMAPADD)

REMAPADD is a SFR to specify the 4 Kbyte area to be remapped.

Address: 0xF0A0 (REMAPADD)
Access: R/W
Access size: 8 bit
Initial value: *1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								REMAPADD							
Bit	-	-	-	-	-	-	-	-	-	RES2	RES1	RES0	REA15	REA14	REA13	REA12
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1

For example, when writing "0x1" to RES2-0 and "0xF" to REA15-12, then remapping them, the area of 0xF000-0xFFFF of code segment 1 is remapped with the area of 0x0000-0x0FFF of segment 0.
A CPU reset for break is happened if setting unused area to this register, and a PC error is not happened.

Bit No.	Bit symbol name	Description
7	-	Reserved bit
6 to 4	RES2 to RES0	Set the code segment of the area to remap. Specify 000b always.
3 to 0	REA15 to REA12	Set the higher 4 bits (bit 15 to 12) of the beginning address of the area to be remapped.

*1: The initial value depends on Code Option settings. See Section 30.2.4 “Code Options 2 (CODEOP2)” for details of Code Option settings.

2.7.4 Reserved register 1

This register is reserved. Don't execute writing this.

Address: 0xF0A4
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								rsvd							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rsvd
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	rsvd	Reserved bit

2.7.5 Reserved register 2

This register is reserved. Don't execute writing this.

Address: 0xF0A6
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								rsvd							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rsvd
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	rsvd	Reserved bit

2.8 Remapping Function

The remapping function replaces the addresses 0x0000 to 0x0FFF (initial boot area) in the program memory space with the specified arbitrary 4 Kbyte area.
Figure 2-4 shows the general scheme of the remapping function.
The program can be started to execute at the area different from the initial boot area using the remapping function, that enables updating(reprograming) the program code area including the initial boot area with the self-programming function.
The remap function enable your application to reprogram the firmware.

Two ways are available to start the remap function.

- Software Remap: Start remapping by resetting only the CPU after setting a remap address into the Flash Remap Address Register (REMAPADD).
- Code Option Remap: Start remapping at the system reset, available by setting the Code Option.

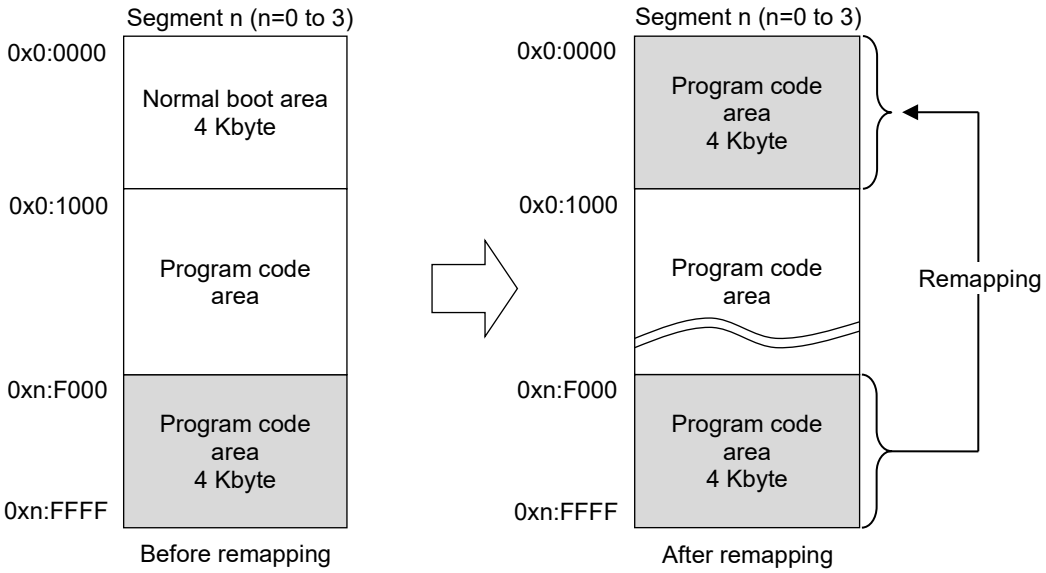


Figure 2-4 Memory Map Before and After Remapping of Program Memory Space

2.8.1 Description of Remapping Function

The remapping function allows the normal boot area of addresses 0x0:0000 to 0x0:0FFF (4 Kbytes) to be replaced (remapped) with the arbitrary 4 Kbyte area set in REMAPADD register.

To use the remapping function, enable it in advance by writing "0" to REMAPMD bit of Code Option 0.

When using the remapping function, the vector table area (reset vector, hardware interrupt vector, and software interrupt vector) is also read from the area specified in REMAPADD register. Prepare the vector table area for areas specified in REMAPADD register.

After remapping, the remapped areas are read through the data segment 0.
If reading the normal boot area (0x0:0000 to 0x0:0FFF) prior to remapping, read it through the data segment 8 in the data memory space (the mirror area of segment 0).

After remapping, if reprogramming the 4Kbyte area in the normal boot area, set "0x0:0000 to 0x0:0FFF" into the Flash Address Register (FLASHA).

2.8.2 Software Remap

The remapping function is activated by software setting a value to REMAPADD register to use the BRK instruction to only reset the CPU.

- Set "0" in advance to REMAPMD bit of Code Option 0 (see Chapter 30 "Code Option" for details on how to set the Code Option).
- Set higher 4 bits of the beginning address of the area to be remapped to REMAPADD register.
- Set ELEVEL of CPU program status word to "2", then execute the BRK instruction (see "nX-U16/100 Core Instruction Manual" for details of ELEVEL and BRK instruction).
- Only the CPU is initialized and it executes the program from the area specified in REMAPADD register.

Figure 2-5 below shows an example of the program script of software remapping.

<If the beginning address of the area to be remapped is 0x0:F000>

```
#asm
    mov     r0,       #00fh
    st      r0,    REMAPADD     ; REMAPADD = 0x0F
    mov     psw, #02h           ; ELEVEL = 2
    nop
    nop
    brk                         ; BRK instruction
#endasm
```

Figure 2-5 Program Script Example of Software Remapping

[Note]

- **If the entire LSI is reset through a system reset, the remapping function is disabled as REMAPADD register is restored with the initial value.**

2.8.3 Code Option Remap

at the system reset on the remap condition.

- If setting both REMAPMD and CREMAPMD to "0", the LSI starts running at the address set in CRES0[*1] and CREA15-CREA12.
- After updating the address in REMAPADD register, the address is not initialized by the CPU reset (BRK instruction) and the remap starts at the updated address. However, REMAPADD register is initialized by the system reset, the LSI starts running at the address specified by the Code Option.

[*1] : In this LSI, always specify "0" for CRES0.

Table 2-7 shows the CPU address at releasing reset of each condition.

Table 2-7 CPU address at releasing reset

Reset	REMAPMD	CREMAPMD	CPU instruction execution start address
CPU reset (BRK instruction)	1	1	0x0000
	1	0	
	0	1	Address set in REMAPADD register
	0	0	
System reset	1	1	0x0000
	1	0	
	0	1	
	0	0	Initial data of REMAPADD register (data set by the Code Options 2)

Chapter 3 Reset Function

3. Reset Function

3.1 General Description

This LSI has a function to reset the CPU, peripheral circuits and other hardware due to the causes described in Table 3-1. This chapter describes the system reset mode, reset input pin reset and power-on reset (POR). See reference chapters for other causes of resets. See Table 3-1 for reference for each cause of resets. See Table 3-2 for the availability of resets for each cause.

Table 3-1 Reference for Details of Causes of Resets

Cause	Reference
Reset input pin reset (pin reset)	This chapter
Power-On Reset (POR)	This chapter
Watchdog timer (WDT) overflow reset	Chapter 10 Watchdog Timer
Watchdog timer (WDT) invalid clear reset	Chapter 10 Watchdog Timer
Low Level Detector reset (LLD reset)	Chapter 22 Voltage Level Supervisor
RAM parity error reset	Chapter 29 Safety Function
Unused ROM area access reset	Chapter 29 Safety Function
CPU reset by BRK instruction execution (when ELEVEL is 2 or higher)	"nX-U16/100 Core Instruction Manual"
Individual reset to the peripheral circuits (Block reset)	Chapter 4 Power Management
One-time reset to the all peripheral circuits and port controller (SOFTR reset)	Chapter 4 Power Management

3.1.1 Features

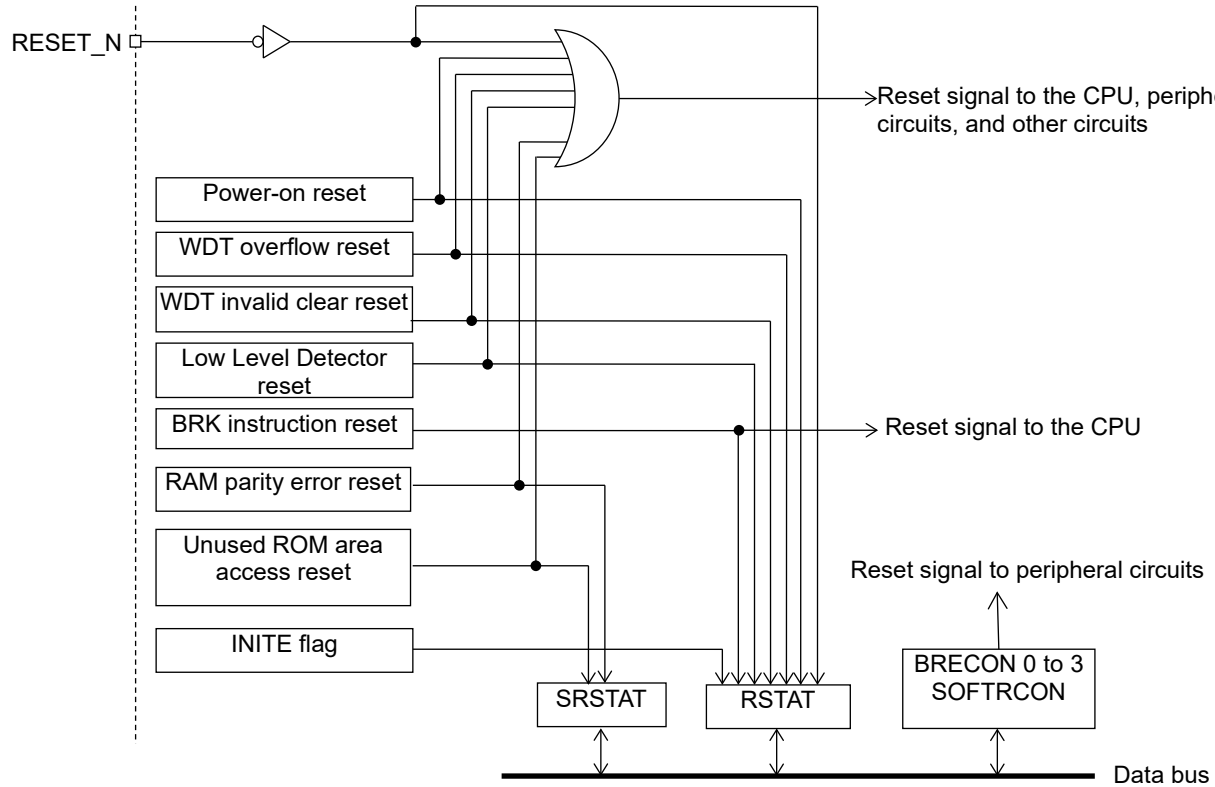
Each reset can uniquely be managed depending on its cause as this function contains following features to identify the cause in an early stage.

- Reset status register (RSTAT) to indicate the cause of the reset
- Reset status register (SRSTAT) to indicate the cause of the safety function reset

In addition, it has the INITE flag function to detect abnormal start-up of the LSI.

3.1.2 Configuration

Figure 3-1 shows the configuration of the reset generation circuit.



RSTAT : Reset status register
SRSTAT : Safety function reset status register
BRECON 0 to 3 : Block reset control register 0 to 3
SOFTRCON : Software reset control register
Other circuits : Power supply and oscillation circuits, start control part, etc.

Figure 3-1 Configuration of Reset Generation Circuit

3.1.3 List of Pins

Pin name	I/O	Function
RESET_N	I	Reset input pin

3.2 Description of Registers

3.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF058	Reset status register	RSTATL	RSTAT	R/W	8/16	Undefined
0xF059		RSTATH		R/W	8	Undefined
0xF05A	Safety function reset status register	SRSTAT	-	R/W	8	Undefined

3.2.2 Reset Status Register (RSTAT)

RSTAT is a SFR to indicate the cause of occurrence of a reset.
When a reset occurs except power-on reset, only the bit that indicates the cause of the reset being set to "1". Other bits (excluding the INITE bit) retain values before occurrence of the reset. When the power-on reset occurs, all bits except POR bit will be "0". After identifying the cause of the reset, write "0xFFFF" to the RSTAT register to initialize the bits of cause of the reset in preparation for the next identification of the cause of the reset.

Address: 0xF058 (RSTATL/RSTAT), 0xF059 (RSTATH)
Access: R/W
Access size: 8/16 bit
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	RSTAT															
Byte	RSTATH								RSTATL							
Bit	-	-	-	-	-	-	-	BRKR	INITE	RSTR	-	LLDR	WDTWR	WDTR	-	POR
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0/1	0	0/1	0	0/1	0/1	0/1	0	0/1

Common description of each bits except bit 7:
It indicates that target reset has occurred. It is initialized to “0” when “1” is written.
0: No target reset occurred. (Initial value)
1: Target reset occurred

Bit No.	Bit symbol name	Description (target reset)
15 to 9	-	Reserved bits
8	BRKR	CPU reset by BRK instruction
7	INITE	A read-only bit to indicate that an abnormality occurred in starting LSI. If this bit is set to "1", restart the LSI by causing a reset to occur with the reset input pin reset, WDT invalid reset, WDT overflow reset or power-on. 0: LSI started-up normally 1: Abnormality occurred in start-up of LSI
6	RSTR	Reset input pin reset
5	-	Reserved bit
4	LLDR	Low Level Detector reset
3	WDTWR	WDT invalid clear reset
2	WDTR	WDT overflow reset
1	-	Reserved bit
0	POR	Power-on reset or command reset of the on-chip debug function.

3.2.3 Safety Function Reset Status Register (SRSTAT)

SRSTAT is a SFR to indicate the cause of occurrence of a safety function reset.
When the safety function reset occurs, only the bit that indicates the cause of the reset occurred is set to "1". Other bits retain values before occurrence of the reset. After identifying the cause of the reset, write "0xFF" to the SRSTAT register to initialize it to "0x00" for preparing the next reset.
See Chapter 29 "Safety Function" for details of the safety function.

Address: 0xF05A (SRSTAT)
Access: R/W
Access size: 8 bit
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SRSTAT							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RPER	FIAR
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Common description of each bits except bit 7:
It indicates that target reset has occurred. It is initialized to “0” when “1” is written.
0: No target reset occurred. (Initial value)
1: Target reset occurred

Bit No.	Bit symbol name	Description (target reset)
7 to 2	-	Reserved bits
1	RPER	RAM parity error reset
0	FIAR	Unused ROM area access reset

3.3 Description of Operation

3.3.1 Operation of Reset Function

Table 3-2 shows the availability of resets for each cause.

Table 3-2 Availability of Resets for Each Cause

Category	Cause	CPU	RAM	Voltage Level Supervisor	Other Peripheral Circuit	System Circuit *1
System reset	Reset input pin reset (pin reset)	●	-	●	●	●
	Power-on reset (POR)	●	-	●	●	●
	WDT overflow reset	●	-	-	●	●
	WDT invalid clear reset	●	-	-	●	●
	Low Level Detector reset	●	-	-	●	●
	RAM parity error reset	●	-	-	●	●
	Unused ROM area access reset	●	-	-	●	●
	Command reset in On-chip debug	●	-	-	●	●
CPU reset	BRK instruction reset	●	-	-	-	-
Peripheral reset	Block reset	-	-	-	●	-
	SOFTR reset	-	-	-	●	-

●: Reset available -: Reset not available
*1: Power circuit, internal oscillation circuit, start control part, code option control part, etc.

[Note]

- **The BRK instruction reset only initializes the CPU if ELEVEL is 2 or higher. Peripheral circuits and other circuits are not initialized. Use the pin reset or the watchdog timer (WDT) reset to surely initialize the LSI when an abnormality is detected.**
- **Command reset in on-chip debug does not reset to Low Level Detector parts. Do initialization of these functions by writing SFRs on debug, if needed. See Chapter 28 for details.**

3.3.2 System Reset Mode

The mode of this LSI is transferred to the system reset mode when a reset occurs by any causes, except for resets caused by the block control register (BRECON 0 to 3) and the software reset control register (SOFTRCON) as well as a CPU reset by the BRK instruction.
The transition to the system reset mode has the highest priority over any other processing. Thus any process in progress up until then will be aborted.
In the system reset mode, the following processes are performed.

1. The fundamental hardware for the LSI operation, such as the power supply circuit and oscillation circuit, is initialized. In addition, functions chosen by the code option are configured. The INITE bit of the reset status register (RSTAT) is set to "1" if an abnormality occurs during the initialization and configuration.
   See the Chapter 30 "Code Option" for details of the code option.
   However, the crystal oscillation circuit and Voltage Level Supervisor are initialized only by power-on reset and reset input pin reset.
2. Peripheral circuits, and special function registers (SFRs) with their initial values defined are initialized. See Appendix A "Registers" and chapters for respective functions for the initial values of the SFRs.
3. The CPU is initialized.
   - All the registers in the CPU are initialized.
   - The contents of addresses 0x0000, 0x0001 in segment 0 of the program memory are set to the stack pointer (SP).
   - The contents of addresses 0x0002, 0x0003 in segment 0 of the program memory are set to the program counter (PC).
4. The transition to the program run mode takes place when the reset is released.

See "nX-U16/100 Core Instruction Manual" for details of registers (SP, PC) in the CPU and the BRK instruction.

[Note]

- **In system reset mode, the contents of data memory (RAM) and SFRs that have an undefined initial value are not initialized. Initialize them by the software.**

3.3.3 Reset Input Pin Reset

Asserting the "L" level to the reset input pin causes the reset state, as well as causing RSTR bit of the reset status register (RSTAT) to be set to "1". Then, negating the reset input pin to the "H" level causes the reset to be released and the program begins to run.
To cause a reset to occur, assert the "L" level which is longer than the reset activation pulse width (P_{RST}).

3.3.4 Power-on Reset

The power-on reset occurs when the power (V_{DD}) is turned on, or when the V_{DD} decreases and stay below the power-on reset trigger voltage (V_{PORF}) for the power-on reset reaction time (P_{POR}). If the power-on reset occurs, POR bit of the reset status register (RSTAT) is set to "1".
When the V_{DD} reaches the power-on reset threshold voltage (V_{PORR}) or above, the reset is released and the CPU starts to run with low-speed clock.
See the data sheet of respective products for power-on reset specifications.

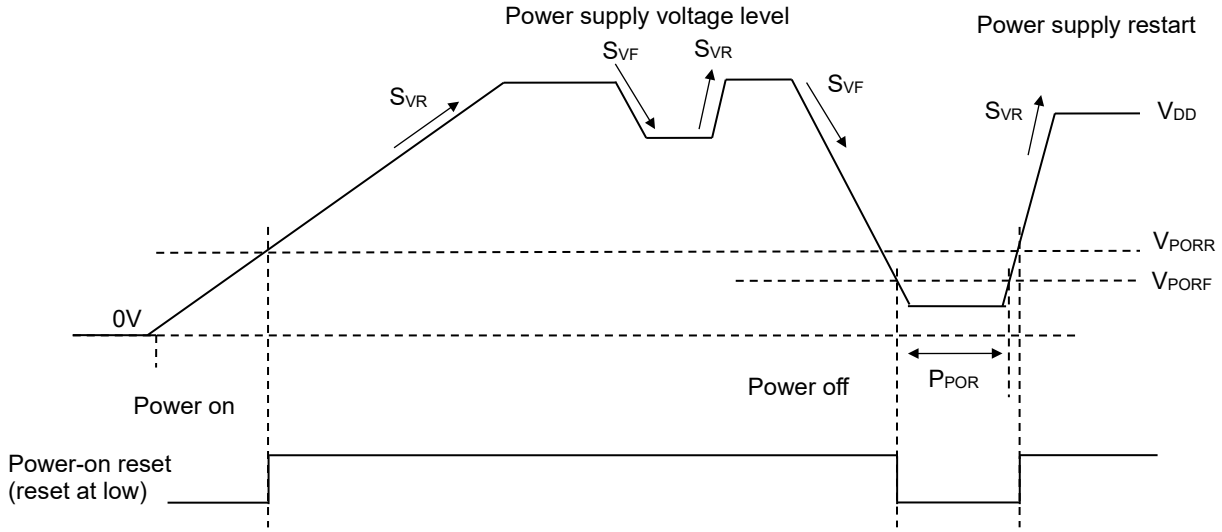


Figure 3-2 Power-on Reset Operation Waveforms

[Note]

- **In case of instantaneous power failure and a pulse shorter than the power-on reset reaction time is asserted to V_{DD}, MCU may not get reset and it may malfunction. In that case, please have preventive measures such as using bypass capacitor to avoid the instantaneous voltage drop or using pin reset to initialize MCU.**

Chapter 4 Power Management

4. Power Management

4.1 General Description

This LSI has two power management modes and block control function to save the current consumption.
The block control function is to control clock supply and reset with respect to each peripheral.

Figure 4-1 shows the general scheme of the regulator.
The regulator generates a constant internal logic voltage (V_{DDL}) independent of the variation of V_{DD} (4.5 V to 5.5 V) using an amplifier for the low power consumption. The V_{DDL} generated by the regulator is supplied to peripheral circuits such as the internal logic circuit, flash memory, RAM, and oscillation circuit.
In order to stabilize the V_{DDL}, connect the VDDL pin to VSS via a capacitor (1µF).

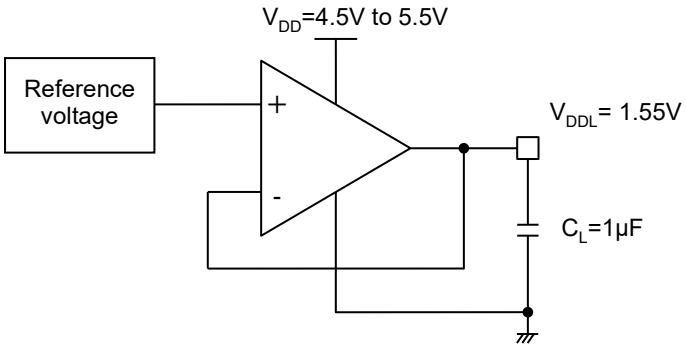


Figure 4-1 General Scheme of Regulator

4.1.1 Features

- 2 standby modes
  - HALT mode : The CPU stops executing instruction, peripheral circuits continue working.
  - STOP mode : The CPU stops executing instruction and all internal clocks stop.
- Stop code acceptor qualifies for entering STOP mode.
- Data of RAM and SFR are retained even in all standby modes.
- Clock supply is control-able peripheral by peripheral to reduce the current consumption, by block clock control registers.
- Reset is control-able peripheral by peripheral by block reset control registers

4.1.2 Configuration

Figure 4-2 shows the transition diagram of the operating state.
The bit symbols in the figure are assigned to the standby control register (SBYCON).

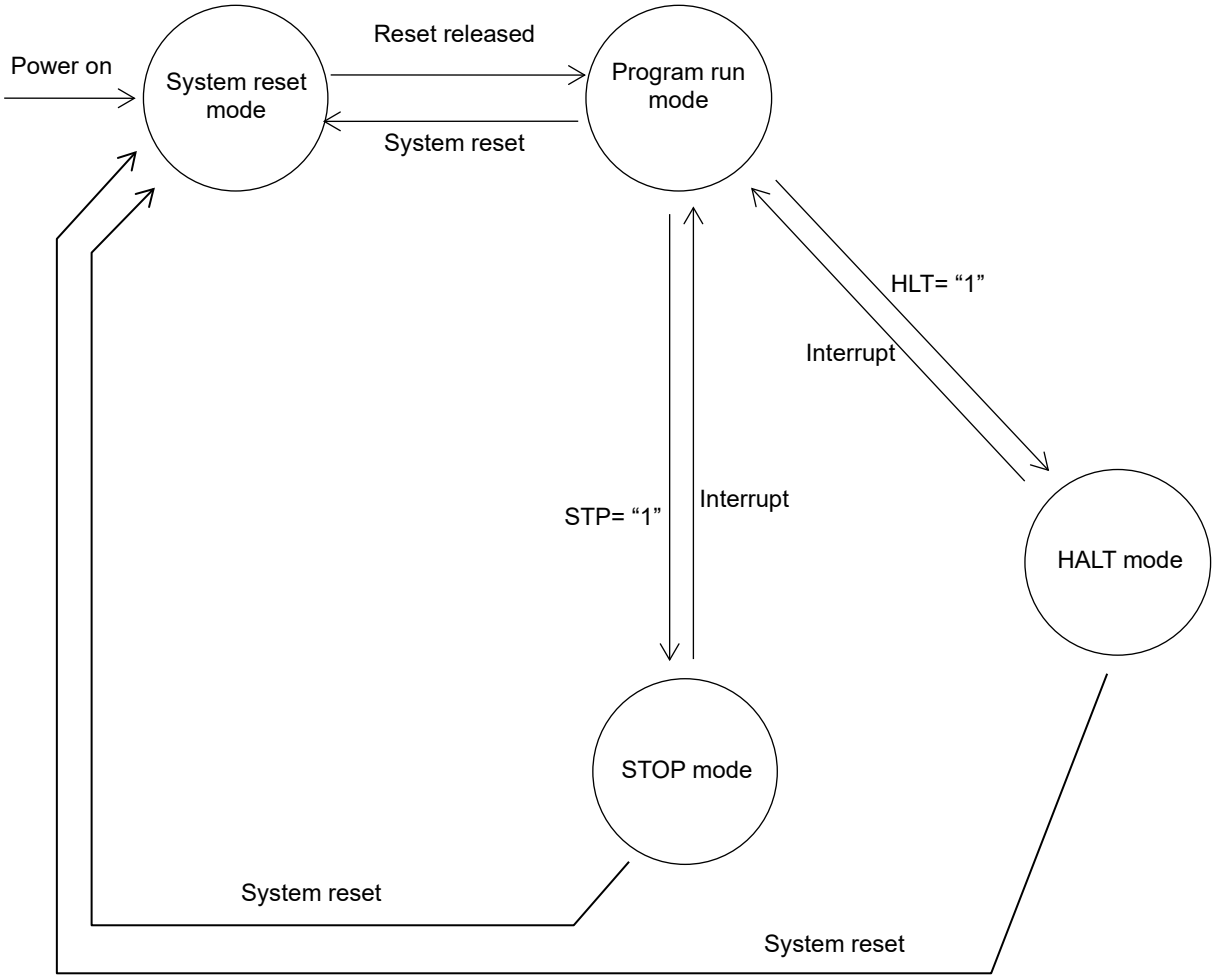


Figure 4-2 Operating State Transition Diagram

Figure 4-3 shows the configuration of the internal power supply.

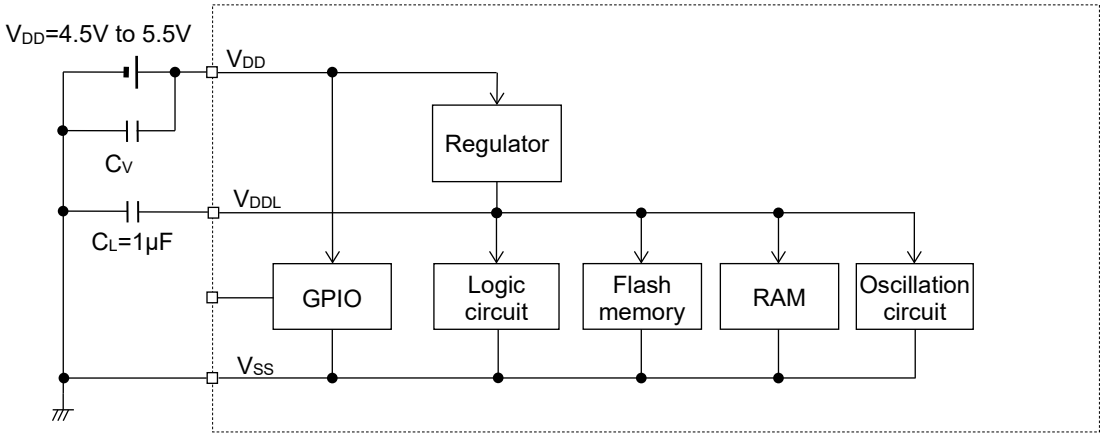


Figure 4-3 Internal Power Supply Configuration

4.1.3 List of Pins

In order to stabilize V_{DDL}, connect the VDDL pin to VSS via a capacitor (1μF).

Pin name	I/O	Function
VDDL	-	Positive power supply for the internal logic circuits

[Note]

- **In order to improve the noise resistance, place the inter-power supply bypass capacitor (C_V) and the internal logic voltage (V_{DDL}) capacitor (C_L : 1μF) in the vicinity of LSI on the user board using the shortest possible wiring without passing through via holes.**

4.2 Description of Registers

4.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF018	Stop code acceptor	STPACP	-	W	8	0x00
0xF019	Reserved	-	-	-	-	-
0xF01A	Standby control register	SBYCONL	SBYCON	W	8	0x00
0xF01B		-		-	-	-
0xF01C	Standby prohibition flag register	SBYEFLG	-	R	8	0x00
0xF01D	Reserved	-	-	-	-	-
0xF05C	Software reset acceptor	SOFTRACP	-	W	8	0x00
0xF05D	Reserved	-	-	-	-	-
0xF05E	Software reset control register	SOFTRCON	-	R/W	8	0x00
0xF05F	Reserved	-	-	-	-	-
0xF070	Block clock control register 0	BCKCON0L	BCKCON0	R/W	8/16	0x00
0xF071		BCKCON0H		R/W	8	0x00
0xF072	Block clock control register 1	BCKCON1L	BCKCON1	R/W	8/16	0x00
0xF073		BCKCON1H		R/W	8	0x00
0xF074	Block clock control register 2	BCKCON2L	BCKCON2	R/W	8/16	0x00
0xF075		BCKCON2H		R/W	8	0x00
0xF076	Block clock control register 3	BCKCON3L	BCKCON3	R/W	8/16	0x00
0xF077		BCKCON3H		R/W	8	0x00
0xF078	Block reset control register 0	BRECON0L	BRECON0	R/W	8/16	0x00
0xF079		BRECON0H		R/W	8	0x00
0xF07A	Block reset control register 1	BRECON1L	BRECON1	R/W	8/16	0x00
0xF07B		BRECON1H		R/W	8	0x00
0xF07C	Block reset control register 2	BRECON2L	BRECON2	R/W	8/16	0x00
0xF07D		BRECON2H		R/W	8	0x00
0xF07E	Block reset control register 3	BRECON3L	BRECON3	R/W	8/16	0x00
0xF07F		BRECON3H		R/W	8	0x00

Table 4-1 shows Availability list of the SFR bit symbols.

Table 4-1 Availability of the SFR bit symbols in BCLCONn register and BRECONn register

●: available, - : unavailable

Control register / bit				Available/ Unavailable
Word symbol	Bit symbol	Word symbol	Bit symbol	
BCKCON0	DCKTM0	BRECON0	RSETM0	●
BCKCON1	DCKOTM0	BRECON1	RSEOTM0	●
	DCKOTM1		RSEOTM1	●
	DCKOTM2		RSEOTM2	●
	DCKOTM3		RSEOTM3	●
	DCKOTM4		RSEOTM4	●
	DCKOTM5		RSEOTM5	●
	DCKI2CU0		RSEI2CU0	●
BCKCON2	DCKUA0	BRECON2	RSEUA0	●
	DCKUA1		RSEUA1	●
	DCKACC		RSEACC	●
BCKCON3	DCKSAD	BRECON3	RSESAD	●
	DCKDAC0		RSEDAC0	●
	DCKDAC1		RSEDAC1	●
	DCKCMP0		RSECMP0	●
	DCKCMP1		RSECMP1	●
	DCKCMP2		RSECMP2	●
	DCKPGA		RSEPGA	●

4.2.2 Stop Code Acceptor (STPACP)

STPACP is a write-only SFR to change the operating state into the STOP mode. This returns "0x00" for reading.

Address : 0xF018 (STPACP)
Access : W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								STPACP							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

How to enter the STOP mode:

Procedure	How to specify the registers	Description
1	Write "0x5n" and "0xAn" (n=arbitrary in 0-F) in sequence into STPACP register.	Enables to enter the STOP mode only once.
2	Set STP bit of SBYCON register to"1".	STP= “1” : Enter the STOP mode

Any other instructions can be executed between the instruction that writes "0x5n" to STPACP and the instruction that writes "0xAn". However, if write data other than "0xAn" after writing "0x5n", the procedure gets invalid, so need write "0x5n" again.

[Note]

- **Writing to the stop code acceptor is invalid on the condition both interrupts enable bits and interrupt request bits are "1", it will not get enabled for entering to the STOP mode.**

4.2.3 Standby Control Register (SBYCON)

SBYCON is a write-only SFR to switch a standby mode.
This returns "0x0000" for reading.

Address : 0xF01A(SBYCONL/SBYCON)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SBYCON															
Byte	-								SBYCONL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STP	HLT
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the WDT interrupt or an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, each standby mode gets canceled and returns to program run mode.

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1	STP	STP is a bit to change the operating state into the STOP mode. When "1" is written in the STP bit after entering the STOP mode is allowed by using STPACP, the operating state enters the STOP mode.
0	HLT	HLT is a bit to change the operating state into the HALT mode.

[Note]

- **The operating state does not enter the standby mode under some conditions. See “4.3.2.3 Note of entering to the standby mode” for detail conditions.**
- **When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".**
- **Insert two NOP instructions in the next to the instruction of that sets HLT and STP bit to "1". The operation without the two NOP instructions is not guaranteed.**

4.2.4 Standby Prohibition Flag Register (SBYEFLG)

SBYEFLG is a read-only SFR to indicate availability of entering to standby mode.
See section 4.3.2.3 “Note of entering to the standby mode” for condition that each bit becomes to “1”

Address : 0xF01C (SBYEFLG)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SBYEFLG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSTP	FHLT
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is a flag of an entering to a target standby mode
0: Available (Initial value)
1: Prohibited

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	FSTP	STOP mode
0	FHLT	HALT mode

4.2.5 Software Reset Acceptor (SOFTRACP)

SOFTRACP is a write-only SFR to enable writing to SOFTCON register. This returns "0x00" for reading.

Address : 0xF05C (SOFTRACP)
Access : W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SOFTRACP							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

How to reset collectively the peripheral circuits:

Procedure	How to specify the registers	Description
1	Write "0x3n" and "0xCn" (n=arbitrary in 0-F) in sequence into the SOFTRACP register.	Enables SOFTR reset only once.
2	Set SOFTR bit of the SOFTRCON register to "1".	SOFTR reset state.

Any other instructions can be executed between the instruction that writes "0x3n" to SOFTRACP and the instruction that writes "0xCn". However, if write data other than "0xCn" after writing "0x3n", the procedure gets invalid, so need write "0x3n" again.

4.2.6 Software Reset Control Register (SOFTRCON)

SOFTRCON is a SFR to reset collectively the all peripheral circuits belong to BRECONn register (n=0 to 3) and general ports.

Address : 0xF05E (SOFTRCON)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SOFTRCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SOFTR
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	SOFTR	Reset collectively the all peripheral circuits belong to the BRECONn register (n=0-3) and general ports. Setting "1" to the bit resets the all peripheral circuits and general ports. The SOFTR is automatically reset to "0" after the reset is completed, so check "0" before re-configuring the peripheral circuits. Enable the reset by writing the SOFTRACP register before setting the SOFTR bit to "1".

[Note]

- **Do not enter the standby mode when SOFTR bit is "1". Ensure SOFTR bit is "0" before entering the standby mode.**

4.2.7 Block Clock Control Register 0 (BCKCON0)

BCKCON0 is a SFR to control supplying the clock of system, high-speed and low-speed to the peripheral circuits. The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.

Address : 0xF070 (BCKCON0L/BCKCON0), 0xF071 (BCKCON0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON0															
Byte	BCKCON0H								BCKCON0L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCKTM0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured supplying clocks to a target peripheral circuit.
0: Supplied clock to a target peripheral circuit (Initial value)
1: Stop clock to a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 1	-	Reserved bits
0	DCKTM0	16-bit timer 0

4.2.8 Block Clock Control Register 1 (BCKCON1)

BCKCON1 is a SFR to control supplying the clock of system, high-speed and low-speed to the peripheral circuits. The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.

Address : 0xF072 (BCKCON1L/BCKCON1), 0xF073 (BCKCON1H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON1															
Byte	BCKCON1H								BCKCON1L							
Bit	-	-	-	DCKI 2CU0	-	-	-	-	-	-	DCK OTM5	DCK OTM4	DCK OTM3	DCK OTM2	DCK OTM1	DCK OTM0
R/W	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured supplying clocks to a target peripheral circuit.
0: Supplied clock to a target peripheral circuit (Initial value)
1: Stop clock to a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 13	-	Reserved bits
12	DCKI2CU0	I^2C Bus Unit 0
11 to 6	-	Reserved bits
5	DCKOTM5	Operational Timer 5
4	DCKOTM4	Operational Timer 4
3	DCKOTM3	Operational Timer 3
2	DCKOTM2	Operational Timer 2
1	DCKOTM1	Operational Timer 1
0	DCKOTM0	Operational Timer 0

4.2.9 Block Clock Control Register 2 (BCKCON2)

BCKCON2 is a SFR to control supplying the clock of system, high-speed and low-speed to the peripheral circuits. The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.

Address : 0xF074 (BCKCON2L/BCKCON2), 0xF075 (BCKCON2H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON2															
Byte	BCKCON2H								BCKCON2L							
Bit	-	-	DCKACC	-	-	-	-	-	-	-	DCKUA1	DCKUA0	-	-	-	-
R/W	R	R	R/W	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured supplying clocks to a target peripheral circuit.
0: Supplied clock to a target peripheral circuit (Initial value)
1: Stop clock to a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 14	-	Reserved bits
13	DCKACC	Multiplier/Divider An initial value of this bit is “0”.
12 to 6	-	Reserved bits
5	DCKUA1	UART 1
4	DCKUA0	UART 0
3 to 0	-	Reserved bits

[Note]

- **DCKACC bit can be set to "0" when the multiplication/division library "muldivu8.lib" is specified. See a manual of the multiplication/division library for how to use.**

4.2.10 Block Clock Control Register 3 (BCKCON3)

BCKCON3 is a SFR to control supplying the clock of system, high-speed and low-speed to the peripheral circuits. The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.

Address : 0xF076 (BCKCON3L/BCKCON3), 0xF077 (BCKCON3H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON3															
Byte	BCKCON3H								BCKCON3L							
Bit	-	-	-	-	-	-	-	-	-	DCKPGA	DCKCMP2	DCKCMP1	DCKCMP0	DCKDAC1	DCKDAC0	DCKSAD
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured supplying clocks to a target peripheral circuit.
0: Supplied clock to a target peripheral circuit (Initial value)
1: Stop clock to a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 7	-	Reserved bits
6	DCKPGA	PGA
5	DCKCMP2	Analog comparator CH2
4	DCKCMP1	Analog comparator CH1
3	DCKCMP0	Analog comparator CH0
2	DCKDAC1	D/A converter CH1
1	DCKDAC0	D/A converter CH0
0	DCKSAD	SA-ADC

4.2.11 Block Reset Control Register 0 (BRECON0)

BRECON0 is a SFR to control resetting the peripheral circuits.

Address : 0xF078(BRECON0L/BRECON0), 0xF079(BRECON0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON0															
Byte	BRECON0H								BRECON0L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RSETM0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured resetting to a target peripheral circuit.
0: Cancel to reset a target peripheral circuit (Initial value)
1: Remain to reset a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 1	-	Reserved bits
0	RSETM0	16-bit timer 0

4.2.12 Block Reset Control Register 1 (BRECON1)

BRECON1 is a SFR to control resetting the peripheral circuits.

Address : 0xF07A (BRECON1L/BRECON1), 0xF07B (BRECON1H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON1															
Byte	BRECON1H								BRECON1L							
Bit	-	-	-	RSEI2CU0	-	-	-	-	-	-	RSEOTM5	RSEOTM4	RSEOTM3	RSEOTM2	RSEOTM1	RSEOTM0
R/W	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured resetting to a target peripheral circuit.
0: Cancel to reset a target peripheral circuit (Initial value)
1: Remain to reset a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 13	-	Reserved bits
12	RSEI2CU0	I^2C Bus Unit 0
11 to 6	-	Reserved bits
5	RSEOTM5	Operational Timer 5
4	RSEOTM4	Operational Timer 4
3	RSEOTM3	Operational Timer 3
2	RSEOTM2	Operational Timer 2
1	RSEOTM1	Operational Timer 1
0	RSEOTM0	Operational Timer 0

4.2.13 Block Reset Control Register 2 (BRECON2)

BRECON2 is a SFR to control resetting the peripheral circuits.

Address : 0xF07C(BRECON2L/BRECON2), 0xF07D(BRECON2H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON2															
Byte	BRECON2H								BRECON2L							
Bit	-	-	RSEACC	-	-	-	-	-	-	-	RSEUA1	RSEUA0	-	-	-	-
R/W	R	R	R/W	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured resetting to a target peripheral circuit.
0: Cancel to reset a target peripheral circuit (Initial value)
1: Remain to reset a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15, 14	-	Reserved bits
13	RSEACC	Multiplier/Divider An initial value of this bit is “0”.
12 to 6	-	Reserved bits
5	RSEUA1	UART 1
4	RSEUA0	UART 0
3 to 0	-	Reserved bits

[Note]

- **RSEACC bit can be set to "0" when the multiplication/division library "muldivu8.lib" is specified. See a manual of the multiplication/division library for how to use.**

4.2.14 Block Reset Control Register 3 (BRECON3)

BRECON3 is a SFR to control resetting the peripheral circuits.

Address : 0xF07E(BRECON3L/BRECON3), 0xF07F(BRECON3H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON3															
Byte	BRECON3H								BRECON3L							
Bit	-	-	-	-	-	-	-	-	-	RSEPGA	RSECMP2	RSECMP1	RSECMP0	RSEDAC1	RSEDAC0	RSESAD
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured resetting to a target peripheral circuit.
0: Cancel to reset a target peripheral circuit (Initial value)
1: Remain to reset a target peripheral circuit

Bit No.	Bit symbol name	Description (target peripheral)
15 to 7	-	Reserved bits
6	RSEPGA	PGA
5	RSECMP2	Analog comparator CH2
4	RSECMP1	Analog comparator CH1
3	RSECMP0	Analog comparator CH0
2	RSEDAC1	D/A converter CH1
1	RSEDAC0	D/A converter CH0
0	RSESAD	SA-ADC

4.3 Description of Operation

4.3.1 Program Run Mode

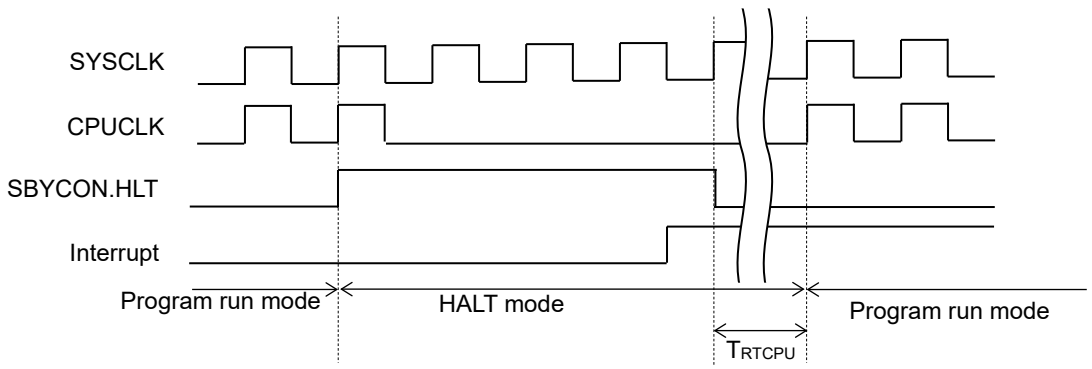
The program run mode is the state where the CPU executes instructions sequentially.
When a reset is released after the reset is generated, the operating state is transferred from the system reset mode to the program run mode.
In addition, if an interrupt request is generated during a standby mode, the mode shifts back to the program run mode.
See Chapter 3 "Reset Function" for the system reset mode.

4.3.2 Standby Mode

4.3.2.1 HALT Mode

The HALT mode is the state where the CPU stops and only the peripheral circuits remain in operation with previous clock condition (LSCLK0 or HSCLK) for the system clock (SYSCLK) selected before entering the HALT mode. See section 4.3.2.5 "Operation of Each Function in Standby Mode" for the operation of each function in the HALT mode.

When "1" is written to the HLT bit of the SBYCON register, the operating state switches the HALT mode.
When a WDT interrupt or an interrupt enabled in registers IE0 to IE7 occurs, the HALT mode is released at the rising edge of the next SYSCLK, then the mode shifts back to the program run mode.
Figure 4-4 shows operation waveforms in the HALT mode.

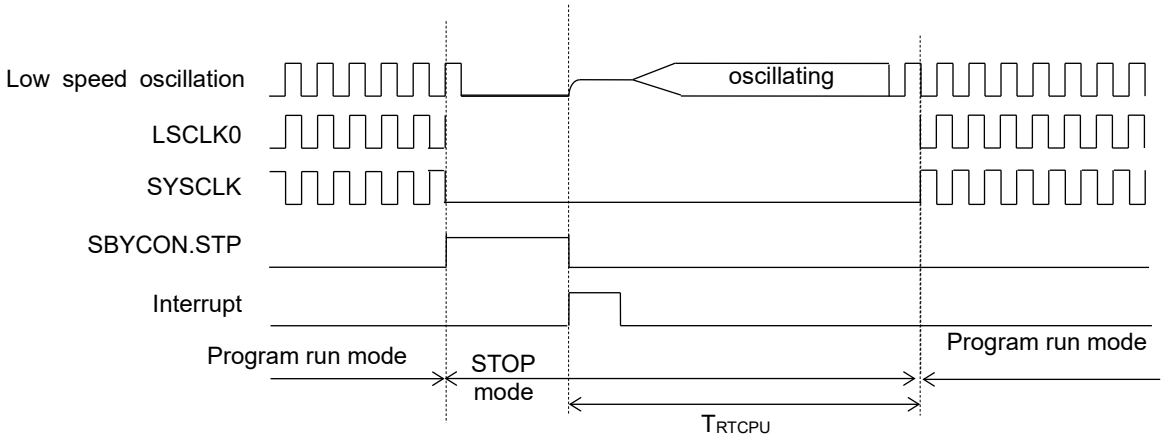


SBYCON : Standby control register
T_{RTCPU} : CPU clock restoring time (see Table 4-6)

Figure 4-4 Operation Waveforms in HALT Mode

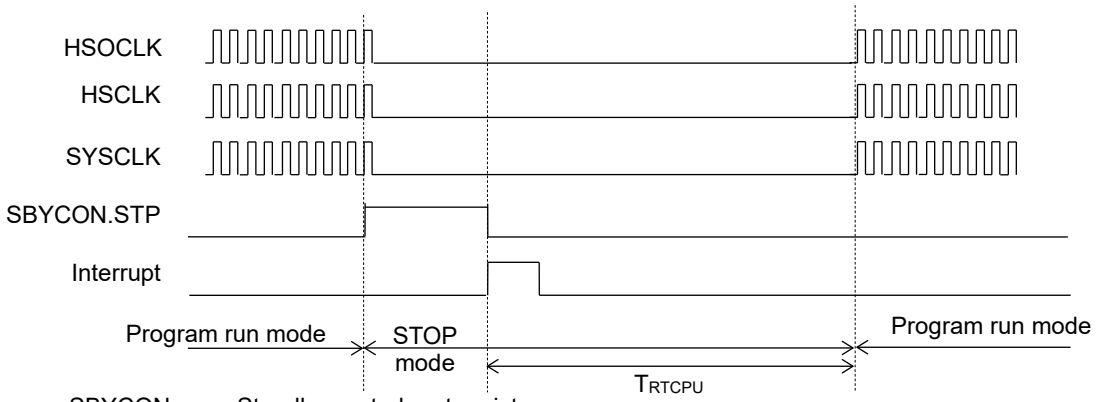
4.3.2.2 STOP Mode

The STOP mode is the state where all clocks are forcibly stopped, and the CPU and the peripheral circuits which need the clock to operate stop. See "4.3.2.5 Operation of Each Function in Standby Mode" for operation of each function in the STOP mode.
To enter the STOP mode, write "0x5n" and "0xAn" (n = arbitrary) in this order to STPACP register to enable the transition to the STOP mode, then write "1" to the STP bit of SBYCON register.
The STOP mode is released by the external interrupts or interrupt requests from the I2C bus unit (slave).
The operating state returns to the program run mode with SYSCLK switched before switching the STOP mode.
Figure 4-5 shows STOP mode operation waveforms.



SBYCON : Standby control part register
T_{RTCPU} : CPU clock wakeup time (see Table 4-6)

Figure 4-5-1 STOP Mode Operation Waveforms of Low-Speed Oscillation Circuit



SBYCON : Standby control part register
T_{RTCPU} : CPU clock wakeup time (see Table 4-6)

Figure 4-5-2 STOP Mode Operation Waveforms of PLL Oscillation Circuit

4.3.2.3 Note of entering to the standby mode

In the following condition, an entering standby mode is canceled, program run mode is continued.
An availability of entering to standby mode is confirmed by monitoring a target bit in SBYEFLG register.
Table 4-2 shows availability of entering to standby mode.

Table 4-2 availability of entering to standby mode (1 : Not available, 0 : Available)

Condition	FSTP	FHLT
When setting some bits of SBYCONL register at the same time.	0	0
When occurring the interrupt request to CPU. Its status is that both interrupt enable register and interrupt request register are asserted.	1	1
When accepter is disabled by SBYACP.	1	0
When A/D conversion of SA-ADC in progress.	1	0
When waiting for stability time of LLD.	1	0
When erasing/programming for data flash memory.	1	0

4.3.2.4 Note on Return Operation from Standby Mode

The operation of returning the standby mode is caused by the interrupt level (ELEVEL) of the program status word (PSW), master interrupt enable flag (MIE), the contents of the register (IE0 to IE7), non-maskable interrupt, or maskable interrupt. The operation varies depending on the cause. See "nX-U16/100 Core Instruction Manual" for details of PSW and Chapter 5 "Interrupts" for IE and IRQ registers respectively. Tables 4-3 shows the return operations from the standby mode for non-maskable interrupt and maskable interrupt respectively.

Table 4-3-1 Return Operation from Standby Mode (for Non-Maskable Interrupt)

ELEVEL	MIE	IEn.m	IRQn.m	Return operation from standby mode
X	X	-	0	Not returned from the standby mode.
3	X	-	1	After returning from the standby mode, the program operation restarts from the instruction next to the instruction that enters the standby mode. The program operation does not go to the interrupt routine.
0,1,2	X		1	After returning from the standby mode, the program operation restarts from the instruction next to the instruction that enters the standby mode. Then the program operation goes to the interrupt routine.

n=0 to 7, m=0 to 7. X: Value-independent

Table 4-3-2 Return Operation from Standby Mode (for Maskable Interrupt)

ELEVEL	MIE	IEn.m	IRQn.m	Return operation from standby mode
X	X	X	0	Not returned from the standby mode.
X	X	0	1	
X	0	1	1	After returning from the standby mode, the program operation restarts from the instruction next to the instruction that enters the standby mode. The program operation does not go to the interrupt routine.
2,3	1	1	1	
0,1	1	1	1	After returning from the standby mode, the program operation restarts from the instruction next to the instruction that enters the standby mode. Then the program operation goes to the interrupt routine.

n=0 to 7, m=0 to 7. X: Value-independent

The ELEVEL of PSW has bits that indicate the state of interrupt process performed by the CPU It is set by the hardware when transferring to the interrupt process or returning from the interrupt.

Table 4-4 State of CPU-Processed Interrupt Indicated by ELEVEL

ELEVEL value	State of CPU-processed interrupt
0	Indicates that the CPU is not processing any interrupt (non-maskable interrupt, maskable interrupt, software interrupt).
1	Indicates that the CPU is processing a maskable or software interrupt.
2	Indicates that the CPU is processing a non-maskable interrupt.
3	Indicates that the CPU is processing an emulator-dedicated interrupt. Usually this is not used in the software.

[Note]

- **Since up to two instructions are executed during the period between the release of standby mode and a transition to interrupt processing, place two NOP instructions next to the instruction set for the standby mode. When a master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", following the execution of the two NOP instructions, the interrupt transition cycle will be executed and execution of the instruction for interrupt routine begins. If MIE is "0", following the execution of the two NOP instructions, the instruction execution is continued from the one that follows the NOP instruction without transition to the interrupt.**

4.3.2.5 Operation of Each Function in Standby Mode

Table 4-5 shows the state of each function block in the standby mode.

Table 4-5 State of Each Function in Standby Mode
● : Operable, - : Not operable

Function blocks	HALT	STOP
Low speed oscillation (Internal RC oscillation)	●	-
High speed oscillation (PLL)	●	-
CPU	-	-
RAM	Retain	Retain
WDT; Watchdog timer	●	-
External interrupt	●	●*1
16-bit timer	●	-*3
Operational timer	●	-*3
UART	●	-
I^2C bus unit (Master)	●	-
I^2C bus unit (Slave)	●	●*2
SA-ADC; Successive approximation type A/D converter	●	-
PGA	●	-
D/A converter*4	●	●
Analog comparator	●	●*1
LLD	●	●*1
BGO operation (erasing/programming for data flash memory)	●	-
Multiplier/Divider	-	-

*1 : If a sampling function is selected, it is forcibly disabled.
*2 : It is available to wake up by coincidence of slave address. A system clock supply is needed for communication after wake up.
*3 : Internal clocks is stop. If external clock is selected, the peripheral circuit operates. However, its operation is not supported.
*4 : Maintains the state before the standby mode transition. Since there is no interrupt notification function, it cannot be used as a interrupt factor.

4.3.2.6 Wake-up Time from Standby Mode

Table 4-6 shows the wake-up time (restoring time) from the standby modes.
See Chapter 6 "Clock Generation Circuit" for details of the FHWUPT register.

Table 4-6 Wake-up Time from Standby Mode (typ.)

Function	Condition	CPU clock restoring time [T_{RTCPU}]	Low-speed clock restoring time (Low-speed RC oscillation) [T_{RTLS}]	High-speed clock restoring time (PLL oscillation) [T_{RTPLL}]	
				FHWUPT=0x01	FHWUPT=0x00
HALT mode	Low-speed CPU clock High-speed clock OFF No CRC calculation	Approx.90µs	Operation continued	Stopped	
	Low-speed CPU clock High-speed clock ON or with CRC calculation	Approx.60µs		Operation continued	
	High-speed CPU clock	-			
STOP mode	Low-speed CPU clock	T_{RTLS}	Approx.3ms	Approx.3ms	Approx.4ms
	High-speed CPU clock	T_{RTPLL}			

[Note]

- **When the FHWUPT register is set to "0x01", the frequency of PLL oscillation clock gradually increases and reaches the target frequency switched by the code option before approx. 2 ms elapse. The PLL oscillation clock during this time period can be used for the SYSCLK, however, accuracy of the frequency is not guaranteed.**

4.3.3 Block Control Function

This LSI has the block clock control function, which stops clock supply for each peripheral circuit to reduce current consumption, and the block reset control function to reset each peripheral circuit.
When setting each bit of the BCKCONn registers (n=0 to 3) to "1", the clock supply to the corresponding peripheral circuits stops, and the current consumption is reduced.
When setting each bit of the BRECONn registers (n=0 to 3) to "1", the corresponding peripheral circuits are reset and those SFRs are set with initial values.
Table 4-7 shows the list of peripheral circuits controllable with the block control function and control registers.

Table 4-7 List of Peripheral Circuits and Control Registers

Peripheral circuit	Block clock control function		Block reset control function		Software reset function
	SFR word symbol	SFR bit symbol	SFR word symbol	SFR bit symbol	SFR bit symbol
16-bit timer 0	BCKCON0	DCKTM0	BRECON0	RSETM0	SOFTR*1
Operational timer 0	BCKCON1	DCKOTM0	BRECON1	RSEOTM0	
Operational timer 1		DCKOTM1		RSEOTM1	
Operational timer 2		DCKOTM2		RSEOTM2	
Operational timer 3		DCKOTM3		RSEOTM3	
Operational timer 4		DCKOTM4		RSEOTM4	
Operational timer 5		DCKOTM5		RSEOTM5	
I^2C bus unit 0		DCKI2CU0		RSEI2CU0	
UART 0	BCKCON2	DCKUA0	BRECON2	RSEUA0	
UART 1		DCKUA1		RSEUA1	
Multiplier/Divider		DCKACC		RSEACC	
Successive approximation type A/D converter	BCKCON3	DCKSAD	BRECON3	RSESAD	
D/A converterCH0		DCKDAC0		RSEDAC0	
D/A converterCH1		DCKDAC1		RSEDAC1	
Analog comparator CH0		DCKCMP0		RSECMP0	
Analog comparator CH1		DCKCMP1		RSECMP1	
Analog comparator CH2		DCKCMP2		RSECMP2	
PGA		DCKPGA		RSEPGA	

*1 : SOFTR resets SFRs for general purpose ports setting too.

After the system reset is released, operation of each peripheral circuit becomes enabled.
If you want to stop the peripheral circuit, set the bit of BRECONn register to "1" to reset the peripheral circuit, and then set the bit of BCKCONn register to "1" to stop the clock of the peripheral circuit. When using, cancel the reset after the clock is supplied.
Also, setting the bit of BRECONn register to "1" only causes a reset to occur while retaining clock supply, enabling each peripheral circuit to be initialized.
In the state where clock supply to each peripheral circuit is suspended or in the reset state, writing to SFRs of corresponding peripheral circuits is disabled. If the clock supply is only stopped, it is possible to read from the SFR.

[Note]

- **If the clock supply is only stopped without resetting each peripheral circuit using the block control function, it may cause the output levels of the timer and communication pins to be fixed, causing the excess current to flow. Also, in the successive approximation type A/D converter, the circuits may stop their function with the current kept flowing.**

4.3.4 Internal Power Supply Voltage

The internal supply voltage, VDDL, is approximately 1.55 V after power on.

Chapter 5 Interrupts

5. Interrupt

5.1 General Description

This LSI has the non-maskable interrupt, maskable interrupts and the software interrupt (SWI).
For details of each interrupt, see the corresponding Chapters.
See Chapter 29 "Safety Function" for the MCU status interrupt.
See "Table 1-2 Main Function List" to confirm the presence/absence of function in each product.

5.1.1 Features

- Master Interrupt Enable (MIE) flag enables or disables collectively the all maskable interrupts. For more details about MIE, see "nX-U16/100 Core Instruction Manual".
- Each maskable interrupt has the enable flag in the register IE0 to IE7.
- The occurrence of interrupt request is confirmable by checking the request flag in IRQ registers.
- The occurrence of interrupt is maskable by setting each request flag by the software in IRQ registers.
- Four interrupt levels are available for each maskable interrupt.

5.2 Description of Registers

Writing to bits of unequipped interrupt is not available. They return 0x0 for reading. See to Table 5-1 for available interrupt.

5.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF020	Interrupt enable register 01	IE0	IE01	R/W	8/16	0x00
0xF021		IE1		R/W	8	0x00
0xF022	Interrupt enable register 23	IE2	IE23	R/W	8/16	0x00
0xF023		IE3		R/W	8	0x00
0xF024	Interrupt enable register 45	IE4	IE45	R/W	8/16	0x00
0xF025		IE5		R/W	8	0x00
0xF026	Interrupt enable register 67	IE6	IE67	R/W	8/16	0x00
0xF027		IE7		R/W	8	0x00
0xF028	Interrupt request register 01	IRQ0	IRQ01	R/W	8/16	0x00
0xF029		IRQ1		R/W	8	0x00
0xF02A	Interrupt request register 23	IRQ2	IRQ23	R/W	8/16	0x00
0xF02B		IRQ3		R/W	8	0x00
0xF02C	Interrupt request register 45	IRQ4	IRQ45	R/W	8/16	0x00
0xF02D		IRQ5		R/W	8	0x00
0xF02E	Interrupt request register 67	IRQ6	IRQ67	R/W	8/16	0x00
0xF02F		IRQ7		R/W	8	0x00
0xF030	Interrupt level control enable register	ILEN	-	R/W	8	0x00
0xF031	Reserved register	-	-	-	-	-
0xF032	Current interrupt level management register	CIL	-	R/W	8	0x00
0xF033	Interrupt level mask register	MCIL	-	R/W	8	0x00
0xF034	Interrupt level control register 0	ILC00	ILC0	R/W	8/16	0x00
0xF035		ILC01		R/W	8	0x00
0xF036	Interrupt level control register 1	ILC10	ILC1	R/W	8/16	0x00
0xF037		ILC11		R/W	8	0x00
0xF038	Interrupt level control register 2	ILC20	ILC2	R/W	8/16	0x00
0xF039		ILC21		R/W	8	0x00
0xF03A	Interrupt level control register 3	ILC30	ILC3	R/W	8/16	0x00
0xF03B		ILC31		R/W	8	0x00
0xF03C	Interrupt level control register 4	ILC40	ILC4	R/W	8/16	0x00
0xF03D		ILC41		R/W	8	0x00
0xF03E	Interrupt level control register 5	ILC50	ILC5	R/W	8/16	0x00
0xF03F		ILC51		R/W	8	0x00
0xF040	Interrupt level control register 6	ILC60	ILC6	R/W	8/16	0x00
0xF041		ILC61		R/W	8	0x00
0xF042	Interrupt level control register 7	ILC70	ILC7	R/W	8/16	0x00
0xF043		ILC71		R/W	8	0x00

Table 5-1 shows presence/absence of interrupt source in each product.

Table 5-1 List of Interrupt Source

● : available, - : unavailable

Register assignment			Interrupt source	Interrupt source symbol	Available/ Unavailable
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			
IRQ0[0]	-	-	WDT Interrupt	WDTINT	●
-	-	-	-	-	-
IRQ0[6]	IE0[6]	ILC0[13:12]	-	-	-
IRQ0[7]	IE0[7]	ILC0[15:14]	-	-	-
IRQ1[0]	IE1[0]	ILC1[1:0]	External Interrupt 0	EXI0INT	●
IRQ1[1]	IE1[1]	ILC1[3:2]	External Interrupt 1	EXI1INT	●
IRQ1[2]	IE1[2]	ILC1[5:4]	External Interrupt 2	EXI2INT	●
IRQ1[3]	IE1[3]	ILC1[7:6]	External Interrupt 3	EXI3INT	●
IRQ1[4]	IE1[4]	ILC1[9:8]	-	-	-
IRQ1[5]	IE1[5]	ILC1[11:10]	-	-	-
IRQ1[6]	IE1[6]	ILC1[13:12]	-	-	-
IRQ1[7]	IE1[7]	ILC1[15:14]	-	-	-
IRQ2[0]	IE2[0]	ILC2[1:0]	-	-	-
IRQ2[1]	IE2[1]	ILC2[3:2]	-	-	-
IRQ2[2]	IE2[2]	ILC2[5:4]	MCU Status Interrupt	MCSINT	●
IRQ2[3]	IE2[3]	ILC2[7:6]	UART00 Interrupt	UA00INT	●
IRQ2[4]	IE2[4]	ILC2[9:8]	UART01 Interrupt	UA01INT	●
IRQ2[5]	IE2[5]	ILC2[11:10]	-	-	-
IRQ2[6]	IE2[6]	ILC2[13:12]	Successive Approximation type A-D Converter Interrupt (SA-ADC Interrupt)	SADINT	●
IRQ2[7]	IE2[7]	ILC2[15:14]	-	-	-
IRQ3[0]	IE3[0]	ILC3[1:0]	-	-	-
IRQ3[1]	IE3[1]	ILC3[3:2]	-	-	-
IRQ3[2]	IE3[2]	ILC3[5:4]	-	-	-
IRQ3[3]	IE3[3]	ILC3[7:6]	-	-	-
IRQ3[4]	IE3[4]	ILC3[9:8]	Operational Timer 0 Interrupt	OTM0INT	●
IRQ3[5]	IE3[5]	ILC3[11:10]	Operational Timer 1 Interrupt	OTM1INT	●
IRQ3[6]	IE3[6]	ILC3[13:12]	16-bit Timer 0 Interrupt	TM0INT	●
IRQ3[7]	IE3[7]	ILC3[15:14]	-	-	-
IRQ4[0]	IE4[0]	ILC4[1:0]	I^2C Bus Unit 0 Interrupt	I2CU0INT	●
IRQ4[1]	IE4[1]	ILC4[3:2]	UART10 Interrupt	UA10INT	●
IRQ4[2]	IE4[2]	ILC4[5:4]	UART11 Interrupt	UA11INT	●
IRQ4[3]	IE4[3]	ILC4[7:6]	Analog Comparator 0 Interrupt	CMP0INT	●
IRQ4[4]	IE4[4]	ILC4[9:8]	Operational Timer 2 Interrupt	OTM2INT	●
IRQ4[5]	IE4[5]	ILC4[11:10]	Operational Timer 3 Interrupt	OTM3INT	●
IRQ4[6]	IE4[6]	ILC4[13:12]	-	-	-
IRQ4[7]	IE4[7]	ILC4[15:14]	-	-	-
IRQ5[0]	IE5[0]	ILC5[1:0]	-	-	-
IRQ5[1]	IE5[1]	ILC5[3:2]	-	-	-
IRQ5[2]	IE5[2]	ILC5[5:4]	Analog Comparator 1 Interrupt	CMP1INT	●

Register assignment			Interrupt source	Interrupt source symbol	Available/ Unavailable
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			
IRQ5[3]	IE5[3]	ILC5[7:6]	Analog Comparator 2 Interrupt	CMP2INT	●
IRQ5[4]	IE5[4]	ILC5[9:8]	Operational Timer 4 Interrupt	OTM4INT	●
IRQ5[5]	IE5[5]	ILC5[11:10]	Operational Timer 5 Interrupt	OTM5INT	●
IRQ5[6]	IE5[6]	ILC5[13:12]	-	-	-
IRQ5[7]	IE5[7]	ILC5[15:14]	-	-	-
IRQ6[0]	IE6[0]	ILC6[1:0]	-	-	-
IRQ6[1]	IE6[1]	ILC6[3:2]	-	-	-
IRQ6[2]	IE6[2]	ILC6[5:4]	-	-	-
IRQ6[3]	IE6[3]	ILC6[7:6]	-	-	-
IRQ6[4]	IE6[4]	ILC6[9:8]	-	-	-
IRQ6[5]	IE6[5]	ILC6[11:10]	-	-	-
IRQ6[6]	IE6[6]	ILC6[13:12]	-	-	-
IRQ6[7]	IE6[7]	ILC6[15:14]	-	-	-
IRQ7[0]	IE7[0]	ILC7[1:0]	-	-	-
IRQ7[1]	IE7[1]	ILC7[3:2]	16-bit Timer 0 Interrupt*1	TM0INT1	●
IRQ7[2]	IE7[2]	ILC7[5:4]	-	-	-
IRQ7[3]	IE7[3]	ILC7[7:6]	-	-	-
IRQ7[4]	IE7[4]	ILC7[9:8]	-	-	-
IRQ7[5]	IE7[5]	ILC7[11:10]	-	-	-
IRQ7[6]	IE7[6]	ILC7[13:12]	-	-	-
IRQ7[7]	IE7[7]	ILC7[15:14]	-	-	-

*1 : In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0,TM0INT1)
Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9. “16-bit timer 0 Interrupt”)

5.2.2 Interrupt Enable Register 01 (IE01)

IE01 is a SFR to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF020(IE0/IE01), 0xF021(IE1)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE01															
Byte	IE1								IE0							
Bit	-	-	-	-	EPI3	EPI2	EPI1	EPI0	-	rsvd	-	-	-	-	-	-
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured enable/disable a target interrupt
0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

After the interrupt is accepted, the master interrupt enables flag (MIE) of the CPU is reset to "0", however, the each applicable flag of a bit of target interrupt is not reset and remains "1".

Bit No.	Bit symbol name	Description
15 to 12	-	Reserved bits
11	EPI3	external interrupt 3 (EXI3INT)
10	EPI2	external interrupt 2 (EXI2INT)
9	EPI1	external interrupt 1 (EXI1INT)
8	EPI0	external interrupt 0 (EXI0INT)
7	-	Reserved bit
6	rsvd	Reserved bit Always write “0” to this bit.
5 to 0	-	Reserved bits

5.2.3 Interrupt Enable Register 23 (IE23)

IE23 is a SFR to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF022(IE2/IE23), 0xF023(IE3)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE23															
Byte	IE3								IE2							
Bit	-	ETM00	EOTM1	EOTM0	-	-	-	-	-	ESAD	-	EUA01	EUA00	EMCS	-	-
R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured enable/disable a target interrupt
0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

After the interrupt is accepted, the master interrupt enables flag (MIE) of the CPU is reset to "0", however, the each applicable flag of a bit of target interrupt is not reset and remains "1".

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14	ETM00	16-bit Timer 0 interrupt0 (TM0INT0)
13	EOTM1	Operational Timer 1 interrupt (OTM1INT)
12	EOTM0	Operational Timer 0 interrupt (OTM0INT)
11 to 7	-	Reserved bits
6	ESAD	SA-ADC interrupt (SADINT)
5	-	Reserved bit
4	EUA01	UART01 interrupt (UA01INT)
3	EUA00	UART00 interrupt (UA00INT)
2	EMCS	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
1 to 0	-	Reserved bits

[Note]

- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1) Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 “16-bit Timer 0 Interrupt”)**

5.2.4 Interrupt Enable Register 45 (IE45)

IE45 is a SFR to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF024(IE4/IE45), 0xF025(IE5)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE45															
Byte	IE5								IE4							
Bit	-	-	EOTM5	EOTM4	ECMP2	ECMP1	-	-	-	-	EOTM3	EOTM2	ECMP0	EUA11	EUA10	EI2CU0
R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured enable/disable a target interrupt
0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

After the interrupt is accepted, the master interrupt enables flag (MIE) of the CPU is reset to "0", however, the each applicable flag of a bit of target interrupt is not reset and remains "1".

Bit No.	Bit symbol name	Description
15, 14	-	Reserved bits
13	EOTM5	Operational Timer 5 Interrupt (OTM5INT)
12	EOTM4	Operational Timer 4 Interrupt (OTM4INT)
11	ECMP2	Analog Comparator 2 Interrupt (CMP2INT)
10	ECMP1	Analog Comparator 1 Interrupt (CMP1INT)
9 to 6	-	Reserved bits
5	EOTM3	Operational Timer 3 Interrupt (OTM3INT)
4	EOTM2	Operational Timer 2 Interrupt (OTM2INT)
3	ECMP0	Analog Comparator 0 Interrupt (CMP0INT)
2	EUA11	UART11 interrupt (UA11INT)
1	EUA10	UART10 interrupt (UA10INT)
0	EI2CU0	I^2C Bus Unit 0 interrupt (I2CU0INT)

5.2.5 Interrupt Enable Register 67 (IE67)

IE67 is a SFR to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF026(IE6/IE67), 0xF027(IE7)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE67															
Byte	IE7								IE6							
Bit	-	-	-	-	-	-	ETM01	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured enable/disable a target interrupt
0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

After the interrupt is accepted, the master interrupt enables flag (MIE) of the CPU is reset to "0", however, the each applicable flag of a bit of target interrupt is not reset and remains "1".

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9	ETM01	16-bit Timer 0 interrupt1 (TM0INT1)
7 to 0	-	Reserved bits

[Note]
- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1) Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 “16-bit Timer 0 Interrupt”)**

5.2.6 Interrupt Request Register 01 (IRQ01)

IRQ01 is a SFR to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF028(IRQ0/IRQ01), 0xF029(IRQ1)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ01															
Byte	IRQ1								IRQ0							
Bit	-	-	-	-	QPI3	QPI2	QPI1	QPI0	-	rsvd	-	-	-	-	-	QWDT
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is a flag of a target interrupt request
0: Not request a target interrupt (Initial value)
1: Request a target interrupt

QWDT bit of the IRQ01 register becomes "1" when the non-maskable Watch Dog Timer (WDT) interrupt occurs and the CPU goes to the interrupt routine regardless the value of the Master Interrupt Enable flag (MIE bit).
Each request flag of IRQ01 except for the QWDT bit becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 01(IE01) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE01 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ01. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ01 becomes "0" automatically when the interrupt request is accepted by the CPU.

Bit No.	Bit symbol name	Description
15 to 12	-	Reserved bits
11	QPI3	external interrupt 3 (EXI3INT)
10	QPI2	external interrupt 2 (EXI2INT)
9	QPI1	external interrupt 1 (EXI1INT)
8	QPI0	external interrupt 0 (EXI0INT)
7	-	Reserved bit
6	rsvrd	Reserved bit. Always write “0” to this bit.
5 to 1	-	Reserved bits
0	QWDT	external interrupt 7 (EXI7INT)

[Note]
- **When rewrite a specific bit of this register, use a bit symbol. Otherwise, other request bits in the same register might be cleared.**
  See Section 5.3.8 “Writing to IRQ01/IRQ23/IRQ45/IRQ67” for more detail.

5.2.7 Interrupt Request Register 23 (IRQ23)

IRQ23 is a SFR to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF02A(IRQ2/IRQ23), 0xF02B(IRQ3)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ23															
Byte	IRQ3								IRQ2							
Bit	-	QTM00	QOTM1	QOTM0	-	-	-	-	-	QSAD	-	QUA01	QUA00	QMCS	-	-
R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is a flag of a target interrupt request
0: Not request a target interrupt (Initial value)
1: Request a target interrupt

Each request flag of IRQ23 becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 01(IE23) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE23 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ23. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ23 becomes "0" automatically when the interrupt request is accepted by the CPU.

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14	QTM00	16-bit Timer 0 interrupt0 (TM0INT0)
13	QOTM1	Operational Timer 1 interrupt (OTM1INT)
12	QOTM0	Operational Timer 0 interrupt (OTM0INT)
11 to 7	-	Reserved bits
6	QSAD	SA-ADC interrupt (SADINT)
5	-	Reserved bit
4	QUA01	UART01 interrupt (UA01INT)
3	QUA00	UART00 interrupt (UA00INT)
2	QMCS	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
1 to 0	-	Reserved bits

[Note]
- **When rewrite a specific bit of this register, use a bit symbol. Otherwise, other request bits in the same register might be cleared.**
  See Section 5.3.8 “Writing to IRQ01/IRQ23/IRQ45/IRQ67” for more detail.
- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1)**
  Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 “16-bit Timer 0 Interrupt”)

5.2.8 Interrupt Request Register 45 (IRQ45)

IRQ45 is a SFR to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF02C(IRQ4/IRQ45), 0xF02D(IRQ5)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ45															
Byte	IRQ5								IRQ4							
Bit	-	-	QOTM5	QOTM4	QCMP2	QCMP1	-	-	-	-	QOTM3	QOTM2	QCMP0	QUA11	QUA10	QI2CU0
R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is a flag of a target interrupt request
0: Not request a target interrupt (Initial value)
1: Request a target interrupt

Each request flag of IRQ45 becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 01(IE45) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE45 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ45. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ45 becomes "0" automatically when the interrupt request is accepted by the CPU.

Bit No.	Bit symbol name	Description
15 to 14	-	Reserved bits
13	QOTM5	Operational Timer 5 Interrupt (OTM4INT)
12	QOTM4	Operational Timer 4 Interrupt (OTM4INT)
11	QCMP2	Analog Comparator 2 Interrupt (CMP2INT)
10	QCMP1	Analog Comparator 1 Interrupt (CMP1INT)
9 to 6	-	Reserved bits
5	QOTM3	Operational Timer 3 Interrupt (OTM4INT)
4	QOTM2	Operational Timer 2 Interrupt (OTM4INT)
3	QCMP0	Analog Comparator 0 Interrupt (CMP0INT)
2	QUA11	UART11 interrupt (UA11INT)
1	QUA10	UART10 interrupt (UA10INT)
0	QI2CU0	I^2C Bus Unit 0 interrupt (I2CU0INT)

[Note]
When rewrite a specific bit of this register, use a bit symbol. Otherwise, other request bits in the same register might be cleared.
See Section 5.3.8 “Writing to IRQ01/IRQ23/IRQ45/IRQ67” for more detail.

5.2.9 Interrupt Request Register 67 (IRQ67)

IRQ67 is a SFR to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading. See to "Table 5-1 List of Interrupt Source" for available peripherals.

Address: 0xF02E(IRQ6/IRQ67), 0xF02F(IRQ7)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ67															
Byte	IRQ7								IRQ6							
Bit	-	-	-	-	-	-	QTM01	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is a flag of a target interrupt request
0: Not request a target interrupt (Initial value)
1: Request a target interrupt

Each request flag of IRQ45 becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 01(IE45) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE45 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ45. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ45 becomes "0" automatically when the interrupt request is accepted by the CPU.

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9	QTM01	16-bit Timer 0 interrupt1 (TM0INT1)
8 to 0	-	Reserved bits

[Note]
- **When rewrite a specific bit of this register, use a bit symbol. Otherwise, other request bits in the same register might be cleared.**
  See Section 5.3.8 Writing to "IRQ01/IRQ23/IRQ45/IRQ67" for more detail.
- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1)**
  Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 "16-bit Timer 0 Interrupt")

5.2.10 Interrupt Level Control Enable Register (ILEN)

ILEN is a special function register (SFR) to enable or disable the interrupt level control.

Address: 0xF030(ILEN)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								ILEN							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ILE
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 1	-	Reserved bits
0	ILE	Control enable or disable the interrupt level control. 0: Disable the interrupt (Initial value) 1: Enable the interrupt

[Note]

- **Disable the interrupt level control function by resetting ILE bit to "0" after resetting the Interrupt level control register 0 to 7 (ILC0 to ILC7) to "0x0000" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled (IE01 to IE67 registers are "0x00").**
- **Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled (IE0 to IE7 registers are "0") or master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.11 Current Interrupt Level Management Register (CIL)

CIL is a SFR to manage the priority level of the interrupt currently being processed by the CPU.
See the section 5.3.6 "How to program the interrupt process when the interrupt level control is enabled".

Address: 0xF032(CIL)
Access: R/W
Access size: 8 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CIL							
Bit	-	-	-	-	-	-	-	-	CILN	-	-	-	CILM 3	CILM 2	CILM 1	CILM 0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
0: A target interrupt is not being processed (Initial value)
1: A target interrupt is being processed

After maskable or non-maskable interrupts to which the priority levels are specified by the interrupt level control registers (ILC0 to 7) is accepted by the CPU, corresponding bits of CIL are automatically set to "1", indicate the currently processing interrupt level.
Interrupts request to the CPU below the currently processed interrupt level will be disabled.
When the multiple bits are "1" in the CIL, it indicates the CPU is processing the multiple interrupts.

Each bit of CIL is automatically set to "1", so it is required to clear to "0" by the software when the interrupt process has been ended. Clear the bit once by writing an arbitrary data at the last in the interrupt process, which resets a flag of CIL corresponding to the highest level.

Bit No.	Bit symbol name	Description
7	CILN	Non-maskable interrupt
6 to 4	-	Reserved bits
3	CILM3	Maskable interrupt with level 4
2	CILM2	Maskable interrupt with level 3
1	CILM1	Maskable interrupt with level 2
0	CILM0	Maskable interrupt with level 1

5.2.12 Masking Interrupt Level Register (MCIL)

MCIL is a SFR to configure masking interrupt level.
It is writeable only when the interrupt level control is enabled by setting ILE bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF033(MCIL)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCIL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	MCIL 3	MCIL 2	MCIL 1	MCIL 0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interrupt notification to the CPU is suspended if interrupt level specified by the ILC0 to ILC7 registers is equal or less than the level specified in the MCIL register.
The interrupt request is notified by lowering the setting value of MCIL register below the suspended interrupt level.

Common description of each bits:
0: A maskable interrupt of a target interrupt level is being processed (Initial value)
1: A maskable interrupt under a target interrupt level is being processed

Bit No.	Bit symbol name	Description
7 to 4	-	Reserved bits
3	MCIL3	Interrupt level 4
2	MCIL2	Interrupt level 3
1	MCIL1	Interrupt level 2
0	MCIL0	Interrupt level 1

5.2.13 Interrupt Level Control Register 0 (ILC0)

ILC0 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting ILE bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF034(ILC00/ILC0), 0xF035(ILC01)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC0															
Byte	ILC01								ILC00							
Bit	-	-	rsvd	rsvd	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	rsvd	Reserved bits. Always write “0” to these bits.
11 to 0	-	Reserved bits

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.14 Interrupt Level Control Register 1 (ILC1)

ILC1 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting ILE bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF036(ILC10/ILC1), 0xF037(ILC11)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC1															
Byte	ILC11								ILC10							
Bit	-	-	-	-	-	-	-	-	ILPI 3H	ILPI 3L	ILPI 2H	ILPI 2L	ILPI 1H	ILPI 1L	ILPI 0H	ILPI 0L
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7,6	ILPI3H, ILPI3L	external interrupt 3 (EXI3INT)
5,4	ILPI2H, ILPI2L	external interrupt 2 (EXI2INT)
3,2	ILPI1H, ILPI1L	external interrupt 1 (EXI1INT)
1,0	ILPI0H, ILPI0L	external interrupt 0 (EXI0INT)

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.15 Interrupt Level Control Register 2 (ILC2)

ILC2 is a SFR) to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting IEL bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF038(ILC20/ILC2), 0xF039(ILC21)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC2															
Byte	ILC21								ILC20							
Bit	-	-	ILSADH	ILSADL	-	-	ILUA01H	ILUA01L	ILUA00H	ILUA00L	ILMCSH	ILMCSL	-	-	-	-
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	ILSADH, ILSADL	SA-ADC interrupt (SADINT)
11,10	-	Reserved bits
9,8	ILUA01H, ILUA01L	UART01 interrupt (UA01INT)
7,6	ILUA00H, ILUA00L	UART00 interrupt (UA00INT)
5,4	ILMCSH, ILMCSL	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
3 to 0	-	Reserved bits

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.16 Interrupt Level Control Register 3 (ILC3)

ILC3 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting ILE bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF03A(ILC30/ILC3), 0xF03B(ILC31)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC3															
Byte	ILC31								ILC30							
Bit	-	-	ILTM00H	ILTM00L	ILOTM1H	ILOTM1L	ILOTM0H	ILOTM0L	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	ILTM00H, ILTM00L	16-bit Timer 0 interrupt0 (TM0INT0)
11,10	ILOTM1H, ILOTM1L	Operational timer 1 interrupt (OTM1INT)
9,8	ILOTM0H, ILOTM0L	Operational timer 0 interrupt (OTM0INT)
7 to 0	-	Reserved bits

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**
- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1) Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 "16-bit Timer 0 Interrupt")**

5.2.17 Interrupt Level Control Register 4 (ILC4)

ILC4 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting IEL bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF03C(ILC40/ILC4), 0xF03D(ILC41)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC4															
Byte	ILC41								ILC40							
Bit	-	-	-	-	ILOTM3H	ILOTM3L	ILOTM2H	ILOTM2L	ILCMP0H	ILCMP0L	ILUA11H	ILUA11L	ILUA10H	ILUA10L	ILI2CU0H	ILI2CU0L
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15 to12	-	Reserved bits
11,10	ILOTM3H, ILOTM3L	Operational timer 3 interrupt (OTM3INT)
9,8	ILOTM2H, ILOTM2L	Operational timer 2 interrupt (OTM2INT)
7,6	ILCMP0H, ILCMP0L	Analog Comparator 0 Interrupt (CMP0INT)
5,4	ILUA11H, ILUA11L	UART11 interrupt (UA11INT)
3,2	ILUA10H, ILUA10L	UART10 interrupt (UA10INT)
1,0	ILI2CU0H, ILI2CU0L	I^2C Bus Unit 0 interrupt (I2CU0INT)

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.18 Interrupt Level Control Register 5 (ILC5)

ILC5 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting IEL bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF03E(ILC50/ILC5), 0xF03F(ILC51)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC5															
Byte	ILC51								ILC50							
Bit	-	-	-	-	ILOTM5H	ILOTM5L	ILOTM4H	ILOTM4L	ILCMP2H	ILCMP2L	ILCMP1H	ILCMP1L	-	-	-	-
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15 to 12	-	Reserved bits
11,10	ILOTM5H, ILOTM5L	Operational timer 5 interrupt (OTM5INT)
9,8	ILCMP2H, ILCMP2L	Analog Comparator 2 Interrupt (CMP2INT)
7,6	ILCMP1H, ILCMP1L	Analog Comparator 1 Interrupt (CMP1INT)
5,4	ILOTM5H, ILOTM5L	Operational timer 5 interrupt (OTM5INT)
3 to 0	-	Reserved bits

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.19 Interrupt Level Control Register 6 (ILC6)

ILC6 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting ILE bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF040(ILC60/ILC6), 0xF041(ILC61)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC6															
Byte	ILC61								ILC60							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	-	Reserved bits

[Note]

- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.20 Interrupt Level Control Register 7 (ILC7)

ILC7 is a SFR to set the interrupt level for each maskable interrupt source.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
It is writeable only when the interrupt level control is enabled by setting IEL bit of the interrupt level control enable register (ILEN) to "1".

Address: 0xF042(ILC70/ILC7), 0xF043(ILC71)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC7															
Byte	ILC71								ILC70							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	ILTM01H	ILTM01L	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured a target interrupt level.
00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description
15 to 4	-	Reserved bits
3,2	ILTM01H, ILTM01L	16-bit Timer 0 interrupt1 (TM0INT1)
1,0	-	Reserved bits

[Note]
- **Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**
- **In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1) Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 “16-bit Timer 0 Interrupt”)**

5.3 Description of Operation

Enabling/disabling the maskable interrupt can be controlled by the master interrupt enable flag (MIE) of the CPU and each interrupt enable register (IE1 to 7).
A WDT interrupt (WDTINT) is unavailable to disable as it is a non-maskable interrupt.
When interrupt conditions are satisfied, the CPU calls a branching destination address from the vector table determined for each interrupt source and the interrupt transfer cycle starts to branch to the interrupt processing routine.

If multiple interrupts are generated concurrently when the interrupt level control function is disabled, they are processed starting from the interrupt with the highest priority (with a smallest interrupt source number). The lower- priority interrupts (with larger interrupt source numbers) remain pending.
If multiple interrupts are generated concurrently when the interrupt level control function is enabled, they are processed starting from the interrupt with both the highest interrupt level and the highest priority level. The lower- priority interrupts remain pending.

Table 5-2 lists the interrupt sources.
The interrupt vector address is an address of the interrupt vector defined in the program memory. See "nX-U16/100 Core Instruction Manual" for details of the interrupt vector address.

Table 5-2 List of interrupt sources

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External /internal source	Interrupt source	Symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
1(high)	IRQ0[0]	-	-	0x0008	Disabled	Internal	WDT interrupt	WDTINT
2	-	-	-	0x000A	Disabled		-	-
3	IRQ0[6]	IE0[6]	ILC0[13:12]	0x000C	Enabled		-	-
4	IRQ0[7]	IE0[7]	ILC0[15:14]	0x000E	Enabled		-	-
5	IRQ1[0]	IE1[0]	ILC1[1:0]	0x0010	Enabled	External	External interrupt 0	EXI0INT
6	IRQ1[1]	IE1[1]	ILC1[3:2]	0x0012	Enabled		External interrupt 1	EXI1INT
7	IRQ1[2]	IE1[2]	ILC1[5:4]	0x0014	Enabled		External interrupt 2	EXI2INT
8	IRQ1[3]	IE1[3]	ILC1[7:6]	0x0016	Enabled		External interrupt 3	EXI3INT
9	IRQ1[4]	IE1[4]	ILC1[9:8]	0x0018	Enabled		-	-
10	IRQ1[5]	IE1[5]	ILC1[11:10]	0x001A	Enabled		-	-
11	IRQ1[6]	IE1[6]	ILC1[13:12]	0x001C	Enabled		-	-
12	IRQ1[7]	IE1[7]	ILC1[15:14]	0x001E	Enabled		-	-
13	IRQ2[0]	IE2[0]	ILC2[1:0]	0x0020	Enabled	Internal	-	-
14	IRQ2[1]	IE2[1]	ILC2[3:2]	0x0022	Enabled		-	-
15	IRQ2[2]	IE2[2]	ILC2[5:4]	0x0024	Enabled		MCU status interrupt*1	MCSINT
16	IRQ2[3]	IE2[3]	ILC2[7:6]	0x0026	Enabled		UART00 interrupt	UA00INT
17	IRQ2[4]	IE2[4]	ILC2[9:8]	0x0028	Enabled		UART01 interrupt	UA01INT
18	IRQ2[5]	IE2[5]	ILC2[11:10]	0x002A	Enabled		-	-
19	IRQ2[6]	IE2[6]	ILC2[13:12]	0x002C	Enabled		SA-ADC interrupt	SADINT
20	IRQ2[7]	IE2[7]	ILC2[15:14]	0x002E	Enabled		-	-
21	IRQ3[0]	IE3[0]	ILC3[1:0]	0x0030	Enabled	Internal	-	-
22	IRQ3[1]	IE3[1]	ILC3[3:2]	0x0032	Enabled		-	-
23	IRQ3[2]	IE3[2]	ILC3[5:4]	0x0034	Enabled		-	-
24	IRQ3[3]	IE3[3]	ILC3[7:6]	0x0036	Enabled		-	-
25	IRQ3[4]	IE3[4]	ILC3[9:8]	0x0038	Enabled		Operational Timer 0 interrupt	OTM0INT
26	IRQ3[5]	IE3[5]	ILC3[11:10]	0x003A	Enabled		Operational Timer 1 interrupt	OTM1INT
27	IRQ3[6]	IE3[6]	ILC3[13:12]	0x003C	Enabled		16-bit Timer 0 interrupt0	TM0INT0
28	IRQ3[7]	IE3[7]	ILC3[15:14]	0x003E	Enabled		-	-
29	IRQ4[0]	IE4[0]	ILC4[1:0]	0x0040	Enabled	Internal	I^2C bus unit interrupt	I2CU0INT
30	IRQ4[1]	IE4[1]	ILC4[3:2]	0x0042	Enabled		UART10 interrupt	UA10INT
31	IRQ4[2]	IE4[2]	ILC4[5:4]	0x0044	Enabled		UART11 interrupt	UA11INT
32	IRQ4[3]	IE4[3]	ILC4[7:6]	0x0046	Enabled		Analog Comparator 0 Interrupt	CMP0INT
33	IRQ4[4]	IE4[4]	ILC4[9:8]	0x0048	Enabled		Operational timer 2 interrupt	OTM2INT
34	IRQ4[5]	IE4[5]	ILC4[11:10]	0x004A	Enabled		Operational timer 3 interrupt	OTM3INT
35	IRQ4[6]	IE4[6]	ILC4[13:12]	0x004C	Enabled		-	-
36	IRQ4[7]	IE4[7]	ILC4[15:14]	0x004E	Enabled		-	-

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External /internal source	Interrupt source	Symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
37	IRQ5[0]	IE5[0]	ILC5[1:0]	0x0050	Enabled	Internal	-	-
38	IRQ5[1]	IE5[1]	ILC5[3:2]	0x0052	Enabled		-	-
39	IRQ5[2]	IE5[2]	ILC5[5:4]	0x0054	Enabled		Analog Comparator 1 Interrupt	CMP1INT
40	IRQ5[3]	IE5[3]	ILC5[7:6]	0x0056	Enabled		Analog Comparator 2 Interrupt	CMP2INT
41	IRQ5[4]	IE5[4]	ILC5[9:8]	0x0058	Enabled		Operational timer 4 interrupt	OTM4INT
42	IRQ5[5]	IE5[5]	ILC5[11:10]	0x005A	Enabled		Operational timer 5 interrupt	OTM5INT
43	IRQ5[6]	IE5[6]	ILC5[13:12]	0x005C	Enabled		-	-
44	IRQ5[7]	IE5[7]	ILC5[15:14]	0x005E	Enabled		-	-
45	IRQ6[0]	IE6[0]	ILC6[1:0]	0x0060	Enabled	Internal	-	-
46	IRQ6[1]	IE6[1]	ILC6[3:2]	0x0062	Enabled		-	-
47	IRQ6[2]	IE6[2]	ILC6[5:4]	0x0064	Enabled		-	-
48	IRQ6[3]	IE6[3]	ILC6[7:6]	0x0066	Enabled		-	-
49	IRQ6[4]	IE6[4]	ILC6[9:8]	0x0068	Enabled		-	-
50	IRQ6[5]	IE6[5]	ILC6[11:10]	0x006A	Enabled		-	-
51	IRQ6[6]	IE6[6]	ILC6[13:12]	0x006C	Enabled		-	-
52	IRQ6[7]	IE6[7]	ILC6[15:14]	0x006E	Enabled		-	-
53	IRQ7[0]	IE7[0]	ILC7[1:0]	0x0070	Enabled	Internal	-	-
54	IRQ7[1]	IE7[1]	ILC7[3:2]	0x0072	Enabled		16-bit Timer 0 interrupt1 *2	TM0INT1
55	IRQ7[2]	IE7[2]	ILC7[5:4]	0x0074	Enabled		-	-
56	IRQ7[3]	IE7[3]	ILC7[7:6]	0x0076	Enabled		-	-
57	IRQ7[4]	IE7[4]	ILC7[9:8]	0x0078	Enabled		-	-
58	IRQ7[5]	IE7[5]	ILC7[11:10]	0x007A	Enabled		-	-
59	IRQ7[6]	IE7[6]	ILC7[13:12]	0x007C	Enabled		-	-
60(low)	IRQ7[7]	IE7[7]	ILC7[15:14]	0x007E	Enabled		-	-

*1 The MCU status interrupt occurs when the following request is asserted.
- RAM parity error
- Data flash erasing/programming completion

These requests are configurable to enable/disable. See Chapter 29 “Safety Function” for detail.

*2 In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1)
It is forbidden to allow both interrupts at the same time. (See Section 5.3.9 “16-bit Timer 0 Interrupt”)

[Note]

- **The WDT interrupt (WDTINT) is a non-maskable interrupt. If the non-maskable interrupt occurs while an interrupt processing is in progress, abort the interrupt processing and proceed with processing the non-maskable interrupt preferentially regardless of multiple interrupts enabled/disabled.**
- **For failsafe, define unused all interrupt vectors. If an unused interrupt occurs, it may indicate the possibility that the CPU went out of control. It is recommended to cause the WDT overflow reset to occur using the infinite loop to initialize the LSI.**

5.3.1 Maskable Interrupt Processing

When an interrupt is generated with MIE set to "1", the following process is executed by hardware and the CPU goes to the interrupt routine.

1. Save the program counter (PC) in ELR1.
2. Save CSR in ECSR1 (not processed if the program memory size is 64 Kbytes or less).
3. Save PSW in EPSW1.
4. Set ELEVEL of PSW to "1".
5. Reset the MIE flag to "0".
6. Set CSR to "0" (not processed if the program memory size is 64 Kbytes or less).
7. Transfer the value of the interrupt vector address to the program counter (PC).

5.3.2 Non-Maskable Interrupt Processing

When an interrupt occurs, the following process is executed by hardware and the CPU goes to the interrupt routine regardless of the value of MIE.

1. Save the program counter (PC) in ELR2.
2. Save CSR in ECSR2 (not processed if the program memory size is 64 Kbytes or less).
3. Save PSW fin EPSW2.
4. Set ELEVEL of PSW to "2".
5. Set CSR to "0" (not processed if the program memory size is 64 Kbytes or less).
6. Transfer the value of the interrupt vector address to the program counter (PC).

5.3.3 Software Interrupt Processing

The software interrupt is arbitrarily produced in software.
When the SWI instruction is performed within the program, a software interrupt occurs, the following process is performed by hardware, and the CPU goes to the software interrupt routine. The vector table is specified with the SWI instruction.

1. Save the program counter (PC) in ELR1.
2. Save CSR in ECSR1 (not processed if the program memory size is 64 Kbytes or less).
3. Save PSW in EPSW1.
4. Set ELEVEL of PSW to "1".
5. Set the MIE flag to "0".
6. Set CSR to "0" (not processed if the program memory size is 64 Kbytes or less).
7. Transfer the value of the interrupt vector address to the program counter (PC).

See "nX-U16/100 Core Instruction Manual" for MIE, the program counter (PC), ELR1, CSR, ECSR1, PSW, EPSW1, ELEVEL, ELR2, ECSR2, EPSW2 and vector table.

5.3.4 Notes on Interrupt Routine (with Interrupt Level Control Disabled)

Writing "0" to the ILE bit of the interrupt level control enable register (ILEN) causes the interrupt level control to be disabled.
The description below shows notes on each of the following states when the interrupt level control is not in use.

- When the sub routine is called/not called in the interrupt routine while execution of the maskable interrupt is in progress (state A).
- When the sub routine is called/not called in the interrupt routine while execution of a non-maskable interrupt is in progress (state B).

State A: Maskable interrupt is being executed

A-1: When a subroutine is not called in an interrupt routine

A-1-1: When multiple interrupts are disabled
• When the script is written in the assembly language
• Processing immediately after the start of interrupt routine execution
No specific notes.
• Processing at the end of interrupt routine execution
Specify the RTI instruction to return the contents of the ELR register to the PC and those of the EPSW register to PSW.
• When the script is written in C
Define the interrupt routine using the INTERRUPT pragma. Specify "1" in the category field. In this way, appropriate codes are produced through the C compiler.

Example of description: State A-1-1

For assembly language:

```
Intrpt_A-1-1;                ; State of A-1-1
DI                           ; Disable interrupt
  :
  :
  :
RTI                          ; Return PC from ELR
                             ; Return PSW form EPSW
                             ; End of interrupt routine
```

For C language:

```
static void Intrpt_A_1_1(void);
#pragma interrupt Intrpt_A_1_1 0x10 1
static void Intrpt_A_1_1(void)
{
_DI();                  /* Disable interrupt */
  :
}                       /* End of interrupt routine */
```

A-1-2: When multiple interrupts are enabled

- When the script is written in the assembly language
  - Processing immediately after the start of interrupt routine execution
    Specify "PUSH ELR, EPSW" to save the interrupt return address and the PSW status in the stack.
  - Processing at the end of interrupt routine execution
    Specify "POP PC, PSW" instead of the RTI instruction to return the contents of the stack to PC and PSW.
- When the script is written in C
  Define the interrupt routine using the INTERRUPT pragma. Specify "2" in the category field. In this way, appropriate codes are produced through the C compiler.

Example of description: State A-1-2

For assembly language:

```
Intrpt_A-1-2;                ; Start
PUSH ELR, EPSW               ; Save ELR and EPSW at the beginning
  :
  :
EI                           ; Enable interrupt
  :
POP PSW, PC                  ; Return PC from the stack
                             ; Return PSW from the stack
                             ; End of interrupt routine
```

For C language:

```
static void Intrpt_A_1_2(void);
#pragma interrupt Intrpt_A_1_2 0x20 2
static void Intrpt_A_1_2(void)
{
_EI();                  /* Enable interrupt */
  :
}                       /* End of interrupt routine */
```

A-2: When a subroutine is called in an interrupt routine

A-2-1: When multiple interrupts are disabled

- When the script is written in the assembly language
  - Processing immediately after the start of interrupt routine execution
    Specify the "PUSH LR" instruction to save the subroutine return address in the stack.
  - Processing at the end of interrupt routine execution
    Specify "POP LR" immediately before the RTI instruction to return from the interrupt processing after returning the subroutine return address to LR.
- When the script is written in C
  Define the interrupt routine using the INTERRUPT pragma. Specify "1" in the category field. In this way, appropriate codes are produced through the C compiler.

Example of description: State A-2-1

For assembly language:

```
Intrpt_A-2-1;              ; Start
PUSH   LR                  ; Save ELR, EPSW, and LR at
                             the beginning
  :
  :
BL Sub_1                   ; Call subroutine Sub_1
  :
POP LR                     ; Return LR from the stack
RTI                        ; End of interrupt routine
```

```
Sub_1;        ;

RT            ; Return PC from LR
              ; End of subroutine
```

For C language:

```
static void Intrpt_A_2_1(void);
#pragma interrupt Intrpt_A_2_1 0x12 1
static void Intrpt_A_2_1(void)
{
  :
Sub_1();            /* Call subroutine Sub_1 */
  :
}                       /* End of interrupt routine */
```

```
void Sub_1(void)
{
:
:
}           /* End of subroutine */
```

[Note]

- **Do not enable interrupts in a subroutine called from an interrupt routine for which multiple interrupts are disabled. Otherwise, the program may run out of control when multiple interrupts occur.**

A-2-2: When multiple interrupts are enabled

- When the script is written in the assembly language
  - Processing immediately after the start of interrupt routine execution
    Specify "PUSH LR, ELR, EPSW, LR" to save the interrupt return address, the subroutine return address, and the EPSW1 status in the stack.
  - Processing at the end of interrupt routine execution
    Specify "POP PSW, PC, LR", instead of the RTI instruction, to return the saved data of the interrupt return address to PC, the saved data of EPSW1 to PSW, and the saved data of LR to LR.
- When the script is written in C
  Define the interrupt routine using the INTERRUPT pragma. Specify "2" in the category field. In this way, appropriate codes are produced through the C compiler.

Example of description: Status A-2-2

For assembly language:

```
Intrpt_A-2-2;                ; Start
PUSH ELR, EPSW, LR           ; Save ELR, EPSW, and LR at the beginning
  :
EI                           ; Enable interrupt
  :
  :
  :
BL Sub_1                     ; Call subroutine Sub_1
  :
POP PSW, PC, LR              ; Return PC from the stack
                             ; Return PSW from the stack
                             ; Return LR from the stack
                             ; End of interrupt routine
```

```
Sub_1;       ;
DI           ; Disable interrupt
  :            :
EI           ; Enable interrupt
RT           ; Return PC from LR
             ; End of subroutine
```

For C language:

```
static void Intrpt_A_2_2(void);
#pragma interrupt Intrpt_A_2_2 0x22 2
static void Intrpt_A_2_2(void)
{
  :
_EI();                  /* Enable interrupt */
  :
Sub_1();              /* Call subroutine Sub_1 */
  :
}                       /* End of interrupt routine */
```

```
void Sub_1(void)
{
_DI();       /* Disable interrupt */
  :
_EI();       /* Enable interrupt */
}            /* End of subroutine */
```

State B: Non-maskable interrupt is being processed

B-1: When a subroutine is not called in an interrupt routine
• When the script is written in the assembly language
  • Processing immediately after the start of interrupt routine execution
    Specify "PUSH ELR, EPSW" to save the interrupt return address and the PSW status in the stack.
  • Processing at the end of interrupt routine execution
    Specify "POP PSW, PC" to return the contents of the stack to PC and PSW.
• When the script is written in C
  Define the interrupt routine using the INTERRUPT pragma. Specify "2" in the category field. In this way, appropriate codes are produced through the C compiler.

Example of description: Status B-1

For assembly language:

```
Intrpt_B-1;                 ; Status B-1
PUSH ELR, EPSW              ; Save ELR and EPSW at the
                              beginning
  :
  :
POP PSW, PC                 ; Return PC from the stack
                            ; Return PSW from the stack
                            ; Return LR from the stack
                            ; End of interrupt routine
```

For C language:

```
static void Intrpt_B_1(void);
#pragma interrupt Intrpt_B_1 0x08 2
static void Intrpt_B_1(void)
{
  :
}                           /* End of interrupt routine */
```

B-2: When a subroutine is called in an interrupt routine

• When the script is written in the assembly language
  • Processing immediately after the start of interrupt routine execution
    Specify "PUSH ELR, EPSW, LR" to save the interrupt return address, the subroutine return address, and EPSW status in the stack.
  • Processing at the end of interrupt routine execution
    Specify "POP PSW, PC, LR" to return the saved data of the interrupt return address to PC, the saved data of EPSW to PSW, and the saved data of LR to LR.
• Description for C language
  Define the interrupt routine by using INTERRUPT pragma and specify "2" in the category field. The C compiler generates the proper codes.

Example of description: Status B-2

For assembly language:

```
Intrpt_B-2;                ; Start
PUSH ELR, EPSW, LR         ; Save ELR, EPSW, and LR at the beginning
  :
  :
BL Sub_1                   ; Call subroutine Sub_1
  :
POP PSW, PC, LR            ; Return PC from the stack
                           ; Return PSW from the stack
                           ; Return LR from the stack
                           ; End
```

```
Sub_1;

RT                 ; Return PC from LR
                   ; End of subroutine
```

For C language:

```
static void Intrpt_B_2(void);
#pragma interrupt Intrpt_B_2 0x08 2
static void Intrpt_B_2(void)
{
  :
Sub_1();           /* Call subroutine Sub_1 */
  :
}                  /* End of interrupt routine */
```

```
void Sub_1(void)
{
  :
  :
}          /* End of subroutine */
```

5.3.5 Flow Charts When Interrupt Level Control Is Enabled

Figure 5-1 shows flow charts of the software interrupt processing when multiple interrupts are disabled and enabled respectively with the interrupt level control enabled.
When multiple interrupts are enabled, save ELR1, ECSR (not processed for products with 64 Kbytes or less of program memory) and EPSW1 in the stack (RAM) so that they are not overwritten by the multiple interrupt. In addition, the EI and DI instructions enable the execution of multiple interrupts due to a high-level maskable interrupt request while "execution of the target process" is in progress.
If a non-maskable interrupt is occurred while the maskable interrupt is being processed, the transition to non-maskable interrupt takes place regardless of multiple interrupts enabled/disabled and the execution of the EI instruction.

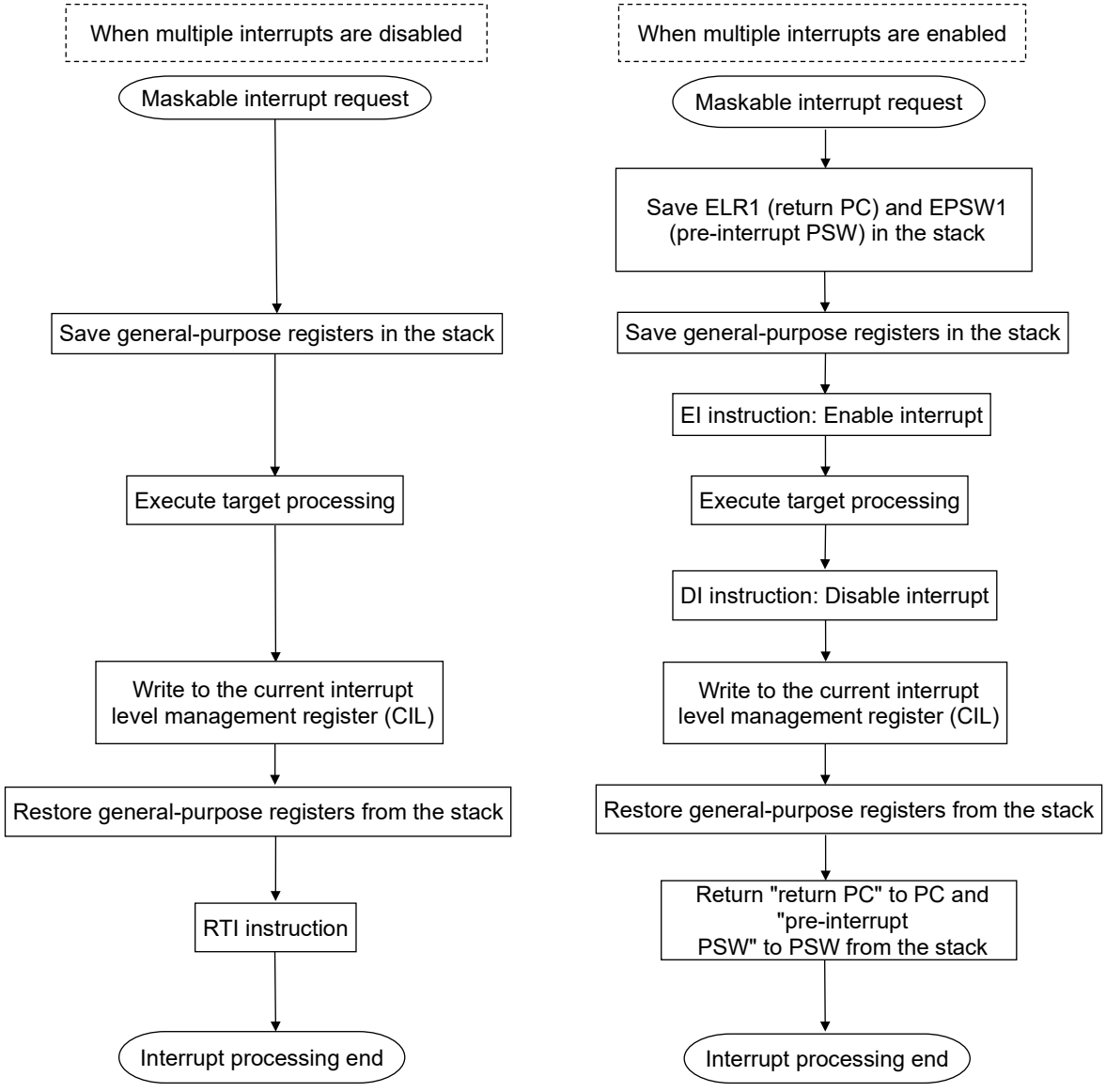


Figure 5-1 Maskable Interrupt Processing Flow

[Note]

- **For processing of non-maskable interrupt, follow the flow chart "When multiple interrupts are enabled". Registers that should be saved in the stack are ELR2 and EPSW2.**
- **When programming in C, it is not required to write program codes for saving/restoring registers because they are generated in the C compiler. However, program codes for enabling/disabling interrupts through EI and DI instructions and for writing to the current interrupt level management register (CIL) must be written. See Section 5.3.6 "How To Write Interrupt Processing When Interrupt Level Control Enabled" for the specific program description.**

5.3.6 How To Write Interrupt Processing When Interrupt Level Control Enabled

This section describes examples of program scripts of interrupt function when ILE of the interrupt level control enable register (ILEN) is set to enable the interrupt level control. See the programming guide of the C compiler for the detailed scripting method of and notes on interrupt processing.

5.3.6.1 Description of Interrupt Function to Disable Multiple Interrupts

To describe the interrupt function to disable multiple interrupts, specify 1 in the category field of the INTERRUPT pragma and SWI pragma. When built-in function _EI is called in the interrupt function to disable multiple interrupts, the C compiler displays an error.
After completion of the target interrupt processing, it is necessary to write to the CIL register and clear the highest current interrupt request level (CILMn bit) to "0". Otherwise, interrupts equivalent to or less than the current interrupt request level is unacceptable.

Example of description

```
static void intr_fn_0A (void);
#pragma interrupt intr_fn_0A 0x0A 1
volatile unsigned short TM1msec;
static void intr_fn_0A (void)
{
        TM1msec++;
        CIL = 0;    /*Clear the highest current interrupt request level*/
}
```

When described as in the example, intr_fn_0A is handled as an interrupt processing function to disable multiple interrupts. the C compiler outputs the assembly code as shown below.

Example of output

```
_intr_fn_0A       :
        push    er0
;;      TM1msec++;
        l       er0,      NEAR _TM1msec
        add     er0,       #1
        st      er0,       NEAR _TM1msec
;;}
;;      CIL = 0;
        mov     r0,        #00h
        st          r0,        0f022h
;;}
        pop     er0
        rti
```

In the interrupt function, the register (here, only ER0) that may be used in the interrupt routine is saved in the stack. "RTI" instruction is used to return from the interrupt function to disable multiple interrupts.
The example below shows how to call other functions from an interrupt function.

Example of description

```
static void intr_fn_10 (void);
#pragma interrupt intr_fn_10 0x10 1
void func (void);
static void intr_fn_10 (void)
{
        func ( );
    CIL = 0;    /*Clear the highest current interrupt request level*/
}
```

Example of output

```
_intr_fn_10     :
    push    lr,     ea
    push    xr0
    l       r0,     DSR
    push    r0

;;  func();
    bl      _func

;;}
;;  CIL = 0;
    mov     r0,     #00h
    st      r0,     0f022h
;;}
    pop     r0
    st      r0,     DSR
    pop     xr0
    pop     ea,     lr
    rti
```

When another function is called from an interrupt function, the output code becomes redundant compared with the case where another function is not called from the interrupt function. Thus the processing time of the interrupt becomes also longer. This is because the C compiler does not know which registers the function func ( ) should use and it save the all registers that may be changed by calling the func ( ) in the stack.

[Note]

- **Do not enable interrupts in a function called from a function for which multiple interrupts are disabled. Otherwise, the program may run out of control when the multiple interrupts occur.**

5.3.6.2 Description of Interrupt Function to Enable Multiple Interrupts

When describing an interrupt function to enable multiple interrupts, specify "2" in the category field in INTERRUPT pragma and SWI pragma. Even if it is not specified in the category field, multiple interrupt are enabled. Built-in function _EI can be called in an interrupt function to enable multiple interrupts.

Example of description

```
static void intr_fn_20 (void);
volatile unsigned short TM2msec;
#pragma interrupt intr_fn_20 0x20 2
static void intr_fn_20 (void)
{
    _EI( );          /*Enable multiple interrupts*/
    TM2msec++;
    _DI( );          /*Disable multiple interrupts*/
    CIL = 0;         /*Clear the highest current interrupt request level*/
}
```

If described as in the example, intr_fn_20 ( ) is handled as an interrupt processing function to enable multiple interrupts. the C compiler outputs the assembly code as shown below.

Example of output

```
_intr_fn_20        :
    push    elr,   epsw
    push    er0

;;      _EI( );          /*Enable multiple interrupts*/
    ei

;;      TM1msec++;
    l       er0,  NEAR _TM2msec
    add     er0,  #1
    st      er0,  NEAR _TM2msec
;;      _DI( );      /*Disable multiple interrupts*/
    di
;;}
;;      CIL = 0;
    mov     r0,       #00h
    st      r0,       0f022h
;;}
    pop     er0
    pop         psw, pc
```

In an interrupt function to enable multiple interrupts, ELR and EPSW are saved in the stack so that they should not be destroyed by multiple interrupts. This is the difference from the interrupt function to disable multiple interrupts. To return from the interrupt function, "POP PSW, PC" is used instead of "RTI".

5.3.7 Interrupt Disable State

The interrupt disable state refers to an operating state where no interrupt is accepted even if the interrupt conditions are satisfied.
The following describes the interrupt disabled state and operation of interrupts in the situation.

State 1. Between the interrupt transfer cycle and the instruction at the beginning of the interrupt routine
When the interrupt conditions are satisfied here, an interrupt is generated immediately after the execution of the instruction at the beginning of the interrupt routine that corresponds to the interrupt already enabled.

State 2. Between the DSR prefix code and the next instruction
When the interrupt conditions are satisfied here, an interrupt is generated immediately after execution of the instruction following the DSR prefix code.

See "nX-U16/100 Core Instruction Manual" for the DSR prefix instruction.

5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67

Use the bit symbol to write to IRQ01/IRQ23/IRQ45/IRQ67 register.
The example below shows how to write "0" to the bit symbol QLTBC0.

Example of description

```
#define clear_bit(n)          ((n)) = 0)

clear bit (QLTBC0);
```

* "n" is the bit symbol name of user's manual.

5.3.9 16-bit timer 0 Interrupt

In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1)
If you want to change the interrupt processing detail according to the conditions, it is possible to do appropriate interrupt processing without determining the condition in the interrupt process by switching the factors to be enabled.
If two places are enabled at the same time, multiple interrupts may occur and unintended operation may occur, so it is prohibited to enable two places at the same time.
Also, if an interrupt occurs with only one of the factors enabled, the IRQ bit of the enabled factor is cleared, but the IRQ bit of the disabled factor remains the interrupt request. To switch the interrupt source, clear the IRQ bit (IRQ3[6]:TM0INT0 and IRQ7[1]:TM0INT1) before switching.

Chapter 6 Clock generation Circuit

6. Clock Generation Circuit

6.1 General Description

The clock generation circuit generates following kinds of clock and supplied them to the CPU or the peripheral circuits.

Table 6-1 Source Clocks

Clock Name	Symbol	Frequency	Description
Internal low-speed clock	RC32K	32.768kHz	Internal generated RC oscillating low-speed clock.
Internal high-speed clock	PLL	64.012288MHz	high-speed clock multiplied LSCLK0. Multiplied by 1953.5

Table 6-2 Clocks generated by the clock generation circuit

Clock Name	Symbol	Frequency	Description
Low speed clock 0	LSCLK0	RC32k	For system and peripheral circuits.
High speed source clock	HSOCLK	PLL	For Operational Timer.
High speed clock	HSCLK	16MHz (HSOCLK/4)	Divided HSOCLK. For system and peripheral circuits.
System clock	SYSCLK	32.768kHz/ 0.5 to 16MHz	A clock. selected from LSCLK0 and a divided HSCLK, for system and communication circuits. It is LSCLK at start-up.
CPU clock	CPUCLK	SYSCLK	A system clock for CPU. It is stop in stand-by mode.
WDT clock	WDTCLK	1.024kHz	For counting watchdog timer.

6.1.1 Features

- Low-speed clock generation circuit
  - Low-speed RC oscillation circuit.
- High-speed clock generation circuit
  - PLL oscillation mode.
  - The wait time at the start of the high-speed clock can be selected.
- WDT clock
  - It is generated by dividing from LSCLK0.

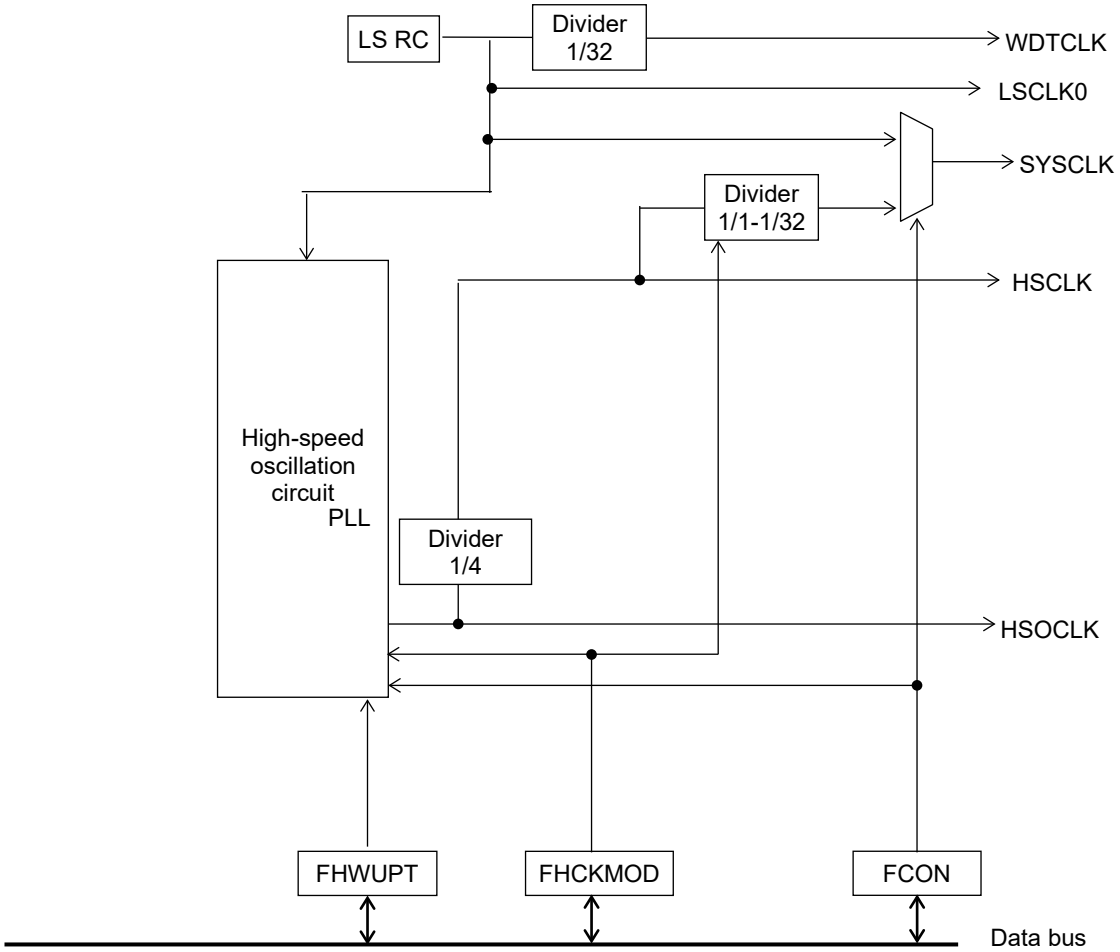
Table 6-3 shows relation of CPU operating mode and PLL frequency.

Table 6-3 Frequency range for high-speed clock operation.

PLL	Maximum operating frequency			output frequency (maximum)
	HSOCLK	HSCLK	SYSCLK	
			Wait mode	
64MHz	64MHz	16MHz	0.5-16MHz	4MHz

6.1.2 Configuration

Figure 6-1 shows the configuration of the clock generation circuit.
Table 6-4 shows the list of operation clocks for each function.



FHCKMOD : High-speed clock mode register
FCON : Clock control register
FHWUPT : High-speed clock wake-up time setting register

Figure 6-1 Configuration of Clock Generation Circuit

[Note]

- **After the power-on or the system reset, LSCLK0 (32.768 kHz) is initially selected as SYSCLK.**

Table 6-4 Operating clock list in each function

Function	SYSCLK	LSCLK0	HSCLK	HSOCLK	WDTCLK
CPU/Data bus	●	-	-	-	-
RAM	●	-	-	-	-
Watchdog timer	-	-	-	-	●
External interrupt control	-	●*1	●*1	-	-
16-bit timer 0-4	-	●	●	-	-
Operational timer	-	●	●	●	-
UART	-	●	●	-	-
I^2C bus unit	●*2	-	-	-	-
SA type A/D converter	-	●	●	-	-
Analog comparator	-	●*1	-	-	-
LLD	-	●*1	-	-	-

● : supplied. - : not supplied.
*1 : for controlling to start or sampling.
*2 : Each block is supplied with a SYSCLK for the register interface. The I^2C bus unit is also used for communication.

6.1.3 List of Pins

This LSI does not have a clock output function.

6.2 Description of Registers

6.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF002	High-speed clock mode register	FHCKMODL	FHCKMOD	R/W	8/16	0x00
0xF003		FHCKMODH		R/W	8	0x03
0xF004	Reserved register	-	-	-	-	-
0xF005	Reserved register	-		-	-	-
0xF006	Clock control register	FCON	FCONW	R/W	8/16	0x00
0xF007		FCON1	-	R/W	8	0x00
0xF008	High-speed clock wake up time setting register	FHWUPT	-	R/W	8	0x00
0xF009	Reserved register	-	-	-	-	-

6.2.2 High-Speed Clock Mode Register (FHCKMOD)

FHCKMOD is a SFR to selects the frequency of the high-speed clock .

Address : 0xF002(FHCKMODL/FHCKMOD), 0xF003(FHCKMODH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0300

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FHCKMOD															
Byte	FHCKMODH								FHCKMODL							
Bit	-	-	-	-	-	SYSC2	SYSC1	SYSC0	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10 to 8	SYSC2 to SYSC0	Select frequency of system clock with high-speed.

SYSC	SYSCLK(16MHz)
000	HSCLK *1
001	1/2 HSCLK *1
010	1/4 HSCLK
011	1/8 HSCLK(Initial value)
100	1/16 HSCLK
101	1/32 HSCLK
110	Do not use (1/32 HSCLK)
111	Do not use (1/32 HSCLK)

7 to 0	-	Reserved bits

Table 6-5 SYSC setting and SYSCLK frequency [MHz]

PLL	SYSC	Frequency [MHz]
64MHz	000	16
	001	8
	010	4
	011	2
	100	1
	101	0.5

6.2.3 Clock Control Register (FCONW)

FCONW is a SFR to control the clock generation circuit.

Address : 0xF006 (FCON/FCONW), 0xF007 (FCON1)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FCONW															
Byte	FCON1								FCON							
Bit	-	-	-	-	-	-	-	-	LPLL	-	-	-	-	-	ENOSC	SELSCLK
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8		Reserved bits
7	LPLL	Indicates that the frequency of the PLL oscillation is within the target error. The LPLL has the read-only attribute. 0: The frequency of PLL oscillation is out of the target error or the PLL oscillation is stopped (Initial value) 1: The frequency of PLL oscillation is within the target error
6 to 2	-	Reserved bits
1	ENOSC	Enable/start or disable/stop the oscillation of the high-speed clock oscillation circuit. 0: Disabled/turn off the high-speed clock oscillation (Initial value) 1: Enabled/turn on the high-speed clock oscillation
0	SELSCLK	Switch the system clock. When ENOSC=0, writing "1" to the SELSCLK bit is disabled. If the FHWUPT register is set to wait for PLL frequency stability (FHUT0 bit = "0"), it is possible to write "1" to the ENOSC bit and the SELSCLK bit at the same time. If you do not wait for frequency stability (FHUT0 bit = "1"), set the ENOSC bit to "1" and set the SELSCLK bit to "1" after executing two NOP instructions. When the high-speed generation circuit is stopped (ENOSC bit = “0”), the SELSCLK bit is fixed to “0”. 0: LSCLK0 (Initial value) 1: High-speed clock chosen by the SYSC2 to SYSC0 bit

6.2.4 High-Speed Clock Wake-up Time Setting Register (FHWUPT)

FHWUPT is a SFR to switch a wakeup time of the high speed clock. This is writable only when ENOSC=0.
See Table 4-6 "Wake-up Time from Standby Mode" in the Chapter 4 for details about the wake-up time from the standby modes.

Address : 0xF008 (FHWUPT)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FHWUPT							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FHUT0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bit
0	FHUT0	Configures the timing of starting to supply the high-speed clock oscillation when ENOSC=1. 0 : After Locked; the frequency clock is stabilized. (Initial value) A system clock stops until the high-speed clock is locked, if SELSCLK is set to “1”. 1 : Soon after ENOSC=1; approx. 30µs. The frequency is not guaranteed as the specification, however it is useable for the system clock. High speed source clock (HSOCLK) supplies the source vibration of the PLL regardless of the setting of this bit.

6.3 Description of Operation

6.3.1 Low-Speed Clock

When the system reset is released, SYSCLK selects a low-speed clock.

6.3.1.1 Low-Speed RC Oscillation Circuit

The low-speed RC oscillation clock is selected for the system clock at the power on.
When a system reset is released, the low-speed clock (LSCLK0) is output and the CPU runs a program after 512 counts of clock chosen for LSCLK0. When the STOP mode is released, the low-speed clock is output after a stable time has elapsed, and the CPU executes the program.
Figure 6-3 shows the configuration of the low-speed RC oscillation circuit.
Figure 6-4 shows the operation waveforms at the start of the low-speed RC oscillation circuit and in the STOP mode.

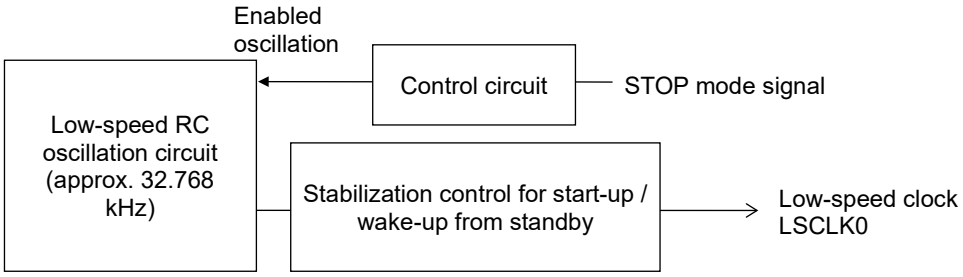


Figure 6-2 Configuration of Low-Speed RC Oscillation Circuit

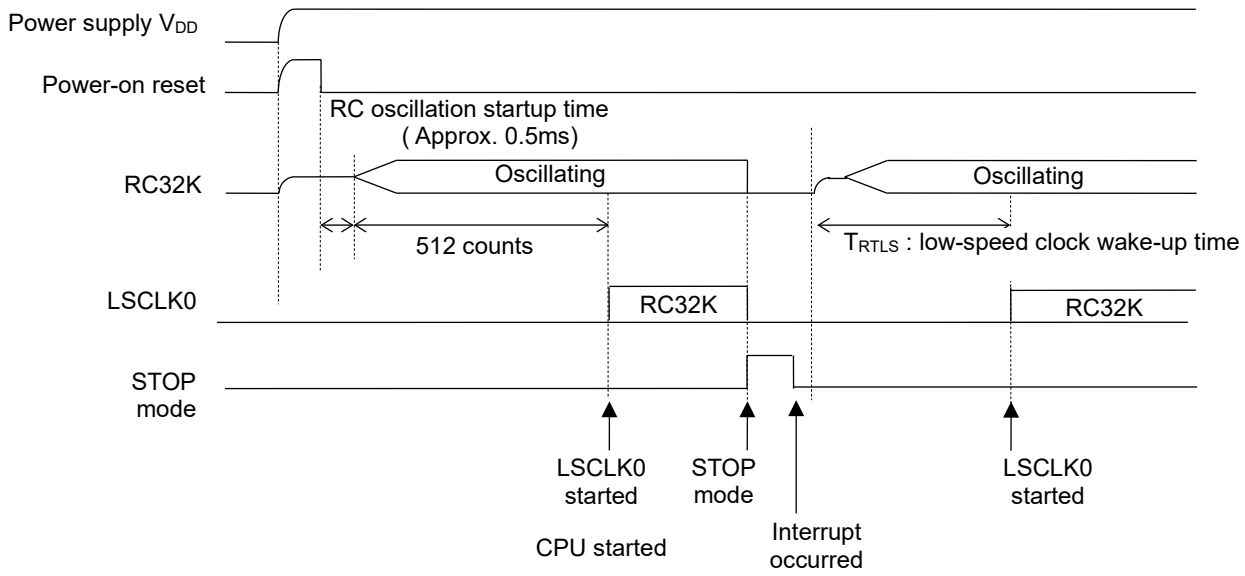


Figure 6-3 Low-speed Clock Operation Waveforms at Start of Low-speed RC Oscillation Circuit and in STOP Mode

6.3.2 High-speed Clock

The PLL oscillation circuit generates the high-speed clock; HSOCLK by multiplying LSCLK0.

6.3.2.1 PLL Oscillation Circuit

The PLL oscillation circuit generates the high-speed clock; HSOCLK by multiplying LSCLK0. The multiplying by 1953.5 is 64MHz of PLL frequency.
After high-speed clock oscillation is enabled, the high-speed clocks; HSCLK is output by continuing count operation until the PLL oscillation clock is stabilized.

When set “1” to FHUT0 bit of FHWUPT register, the clock supply is started approximately 30 μs after the high-speed clock oscillation is enabled.
The clock frequency reaches to the target approximately 1 ms after the high-speed clock oscillation is enabled. Although the frequency within the 1 ms is not guaranteed, it can be used for the system clock.
When FHUT0 bit is set to "1" and the system clock is switched to HSCLK immediately after the PLL oscillation starts, setENOSC bit of the FCONW register to "1" and set SELSCLK bit of FCONW register to "1" after executing the two NOP instructions.
When set “0” to FHUT0 bit, the clock supply is started approximately 1ms after it is enabled.

In addition, the PLL oscillation circuit stops oscillation when entering the STOP mode.
Its oscillation is output, after wakeup from standby and waiting stabilization.

Figure 6-4 shows the PLL oscillation circuit configuration.

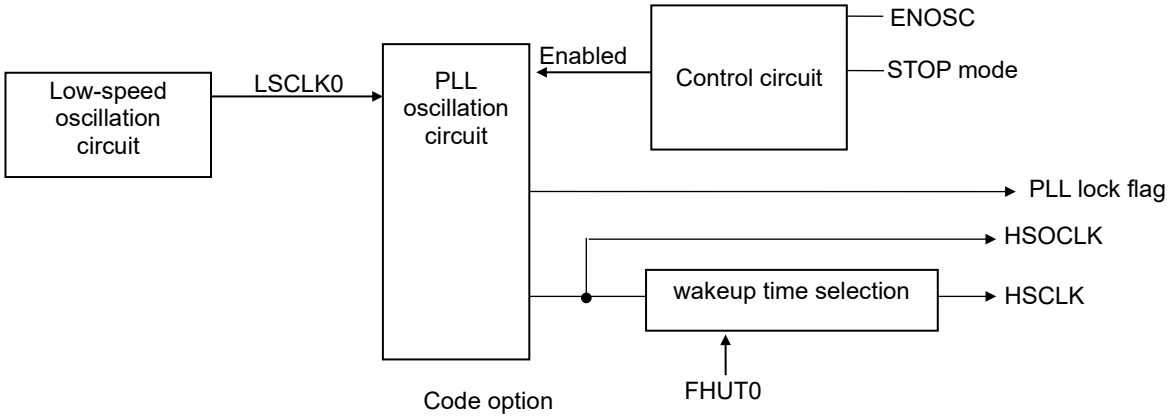


Figure 6-4 PLL Oscillation Circuit Configuration

Figure 6-5 show the high-speed clock operation waveforms at startup PLL circuit, in the standby mode.
See chapter 4 “Power Management” for details of the STOP mode.

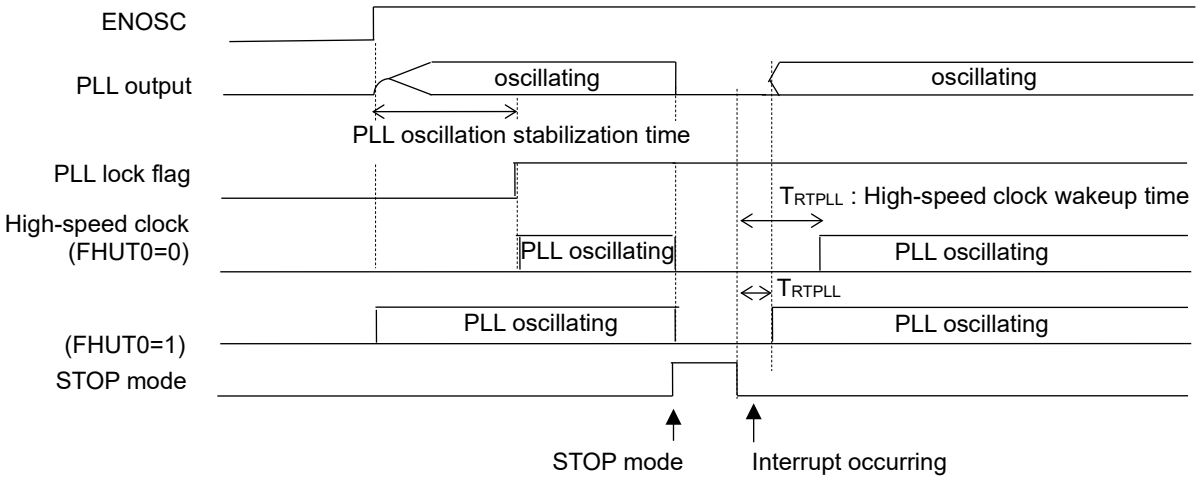


Figure 6-5 High-speed clock operation at startup PLL circuit and in STOP mode

6.3.2.2 High-speed Clock Control

The following is a procedure to control high-speed clock.

In the case of with waiting stabilization for high-speed clock.
(1) Set with waiting stabilization for high-speed clock by FHWUPT register. (FHUT0=0)
(2) Set enabled PLL oscillation by ENOSC=1. SELSCLK bit can be set to “1” at once.

In the case of with waiting stabilization; FHUT0=0: The clock is supplied after LPLL bit becomes to “1”.
The system clock stops until the high-speed clock oscillates when SELSCLK=1.

In the case of with not waiting stabilization for high-speed clock.
(1) Set without waiting stabilization for high-speed clock by FHWUPT register. (FHUT0=1).
(2) Set enabled PLL oscillation by ENOSC=1.
(3) When select HSCLK as the system clock (SELSCLK="1"), executing the two NOP instructions after (2) and then change SELSCLK.

In the case of without waiting stabilization; FHUT0=1: Although the high-speed clock frequency is not guaranteed just after setting ENOSC=1, it can be used for the system clock and peripheral clocks. If specific frequency is required, wait stabilization time; LPLL=1.

In the case of stop the high-speed clock:
Set “0” to ENOSC bit to turn off high-speed clock by the software. Then SELSCLK is cleared at once, and the system clock is switched to low-speed clock.
If high-speed clock is kept on, do not set “0” to ENOSC bit, and set “0” to SELSCLK bit only.

6.3.3 WDT Clock (WDTCLK)

1kHz clock for WDT is oscillation that frequency is 1.024kHz (typ.), which is divided by 32 degrees of RC32K.
It is supplied to the WDT, trigger for an operational timer.
When WDT operation is stopped with the code option, the WDT clock supply is also stopped.
In case of want to use the WDT clock as a trigger in an operational timer, pay attention to the following code option settings.

In the case of the WDT operation is enabled by setting code option; WDTMD=1 :
WDT clock is oscillating after system reset is released. It turns off at entry to STOP mode, and then it turns on at wakeup from standby. In the HALT-D mode, it stops only when WDTPWMD1=0 is selected by setting code option.
The clock wakeup time is approx. 2ms.

In the case of WDTMD = 0 :
WDT clock has stopping after system reset is released.
See Chapter 30 “Code Option” for how to set code options.

Figure 6-6-1 shows WDT clock operation waveform when the RC32K is started up and in STOP mode.
Figure 6-6-2 shows WDTPWMD0=0, Indicates the WDT clock operation waveform in HALT mode.

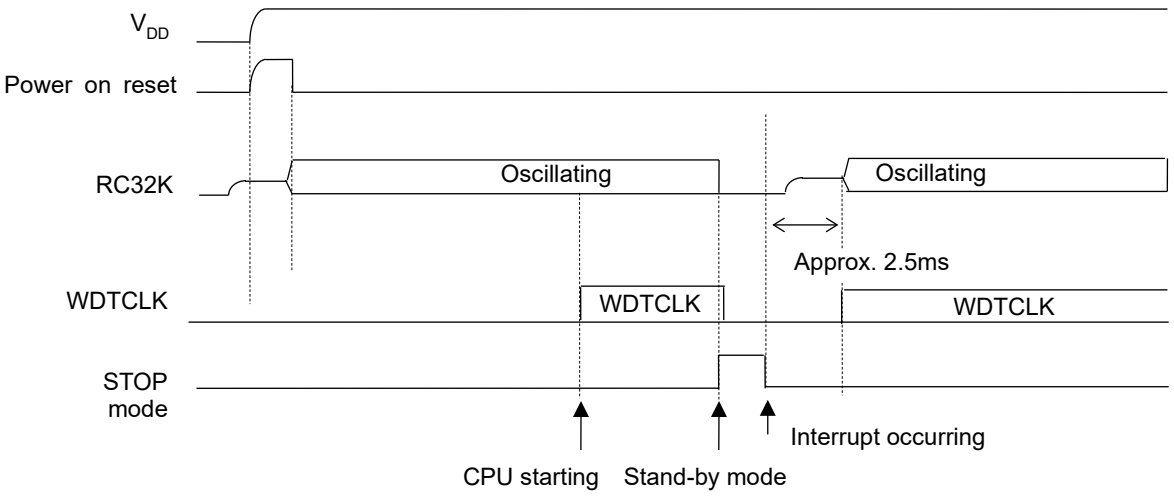


Figure 6-6-1 Startup and Wakeup from STOP mode of WDTCLK

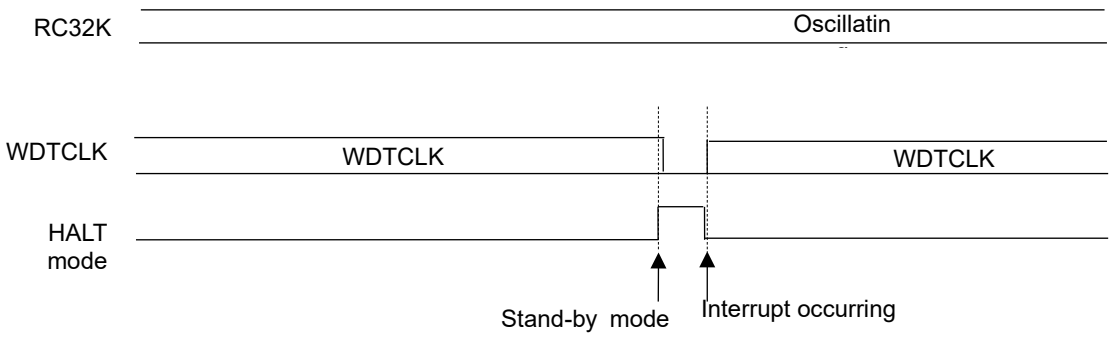


Figure 6-6-2 HALT mode

6.3.4 System Clock

A system clock is LSCLK0 with RC32K after power-on-reset/pin reset.
A system clock can be dynamically switched the high-speed system clock or LSCLK0 by SELSCLK bit. The high-speed system clock can be dynamically changed dividing value by SYSC bits.
There are 2 types as system clock; CPUCLK supplied to CPU and SYSCLK supplied to peripherals.
The CPUCLK stops in all stand-by mode. The SYSCLK normally supplies in the HALT mode.
See chapter 4 “Power Management” for detail.

[Note]

- **While the CPU is running with the low-speed clock, if running the peripheral circuits with the high-speed clock which can frequently generate interrupts, the operation may fail to function properly due to the CPU becoming incapable of processing interrupts in time. If interrupts frequently occur for reasons such as short interrupt cycles of peripheral circuits, take into account the operating frequency of the CPU so that it can process interrupts in time.**
-

Chapter 8 16-Bit Timer

8. 16-Bit Timer

8.1 General Description

The 16-bit timer enables following functions.

- Generate periodical interrupts in an arbitrary period
- Generate one shot interrupts in an arbitrary period

Table 8-1 shows the number of channels.

Table 8-1 Number of 16-bit Timer channels
● : available

Channel no.	Available/Unavailable
0	●

8.1.1 Features

Operation mode	Description
Repeat mode	Count-able up to the max. 0xffff Repeat the specified operation until stop by the software.
One shot mode	Count-able up to the max. 0xffff Run the specified operation once and stop it.

- Selectable counter clock from various sources (divided by 1 to 8 of LSCLK0, HSCLK, external clock, Operational timer triggers)
- A timer interrupt request is generated when the value of the timer counter register value coincides with that of the 16-bit timer n data register

8.1.2 Configuration

Figure 8-1 shows configuration of the 16-bit timer

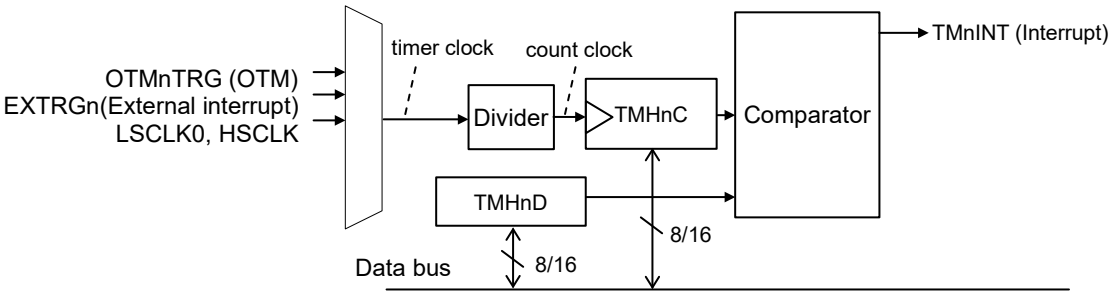


Figure 8-1 Configuration of the timer

TMnINT : 16-bit timer n interrupt request
EXTRGn : EXIn pin input (come through the noise filter of the external interrupt function)
TMHnD : 16-bit timer n data register
TMHnC : 16-bit timer n counter register

8.2 Description of Registers

8.2.1 List of Registers

Registers for unequipped channels are not available to use. They return 0x0000 for reading.

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF300	16-bit timer 0 data register	TMH0DL	TMH0D	R/W	8/16	0xFF
0xF301		TMH0DH		R/W	8	0xFF
0xF302	16-bit timer 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF303		TMH0CH		R/W	8	0x00
0xF304	16-bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF305		TMH0MODH		R/W	8	0x00
0xF306 to 0xF33F	Reserved registers	-	-	-	-	-
0xF340	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF341		TMHSTRH		W	8	0x00
0xF342	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00
0xF343		TMHSTPH		W	8	0x00
0xF344	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF345		TMHSTATH		R	8	0x00

8.2.2 16-Bit Timer n Data Register (TMHnD :n=0)

TMHnD is a SFR to set the comparison value with the 16-bit timer n counter register (TMHnC).
When TMHnD register is rewritten during timer operation, the value written to TMHnD is set at the beginning of the next cycle.
When TMHnD register is rewritten while the timer is stopped, the value is set immediately.

Address : 0xF300 (TMH0DL/TMH0D), 0xF301 (TMH0DH)
Access : R/W
Access size : 8/16 bit
Initial value : 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHnD															
Byte	TMHnDH								TMHnDL							
Bit	THnD15	THnD14	THnD13	THnD12	THnD11	THnD10	THnD9	THnD8	THnD7	THnD6	THnD5	THnD4	THnD3	THnD2	THnD1	THnD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[Note]

- **When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD.**

8.2.3 16-Bit Timer n Counter Register (TMHnC :n=0)

TMHnC is a SFR that functions as a 16-bit binary counter.
This is reset to 0x0000 at the reset function and also when the following event occurred.

- When an arbitrary value is written in this register
- When the value of TMHnD register coincides with that of TMHnC register

Address : 0xF302 (TMH0C/TMH0CL), 0xF303 (TMH0CH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHnC															
Byte	TMHnCH								TMHnCL							
Bit	THnC15	THnC14	THnC13	THnC12	THnC11	THnC10	THnC9	THnC8	THnC7	THnC6	THnC5	THnC4	THnC3	THnC2	THnC1	THnC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This data is counted up synchronizing at the rising edge of the count clock.
Reading value is always available if timer clock source is as same as system clock source.

An available condition:

System clock	Timer clock
LSCLK0	LSCLK0
HSCLK	HSCLK

[Note]

- **Read TMHnC register twice to verify the valid data to prevent reading uncertain data while counting-up, if a source of timer clock is as different as one of system clock.**

8.2.4 16-Bit Timer n Mode Register (TMHnMOD :n=0)

TMHnMOD is a SFR to control the operation mode of 16-bit timer.

Address : 0xF304 (TMH0MODL/TMH0MOD), 0xF305 (TMH0MODH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHnMOD															
Byte	TMHnMODH								TMHnMODL							
Bit	-	-	-	-	-	-	THn OST	-	-	-	THn DIV1	THn DIV0	THnCK 3	THnCK 2	THnC K1	THnCK 0
R/W	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9	TLnOST	Select the operation mode. 0: Repeat mode (Initial value) 1: One-shot mode
8 to 6	-	Reserved bits
5 to 4	THnDIV1 to THnDIV0	Select frequency dividing ratio for the count clock 00: No dividing (Initial value) 01: Divided by 2 10: Divided by 4 11: Divided by 8
3 to 0	THnCK3 to THnCK0	Select the timer clock source. 0000: LSCLK0 (Initial value) 0001: HSCLK 0010: Do not use 0011: Do not use 0100: EXTRG0 0101: EXTRG1 0110: EXTRG2 0111: EXTRG3 1000: OTM0TRG 1001: OTM1TRG 1010: OTM2TRG 1011: OTM3TRG 1100: OTM4TRG 1101: OTM5TRG 1110: Do not use 1111: Do not use

[Note]

- **Set TMHnMOD when the timer n is stopped (THnSTAT bits of TMHSTAT/TMHXSTAT register are "0"). If it is changed while it is operating, the operation is not guaranteed.**

8.2.5 16-Bit Timer Start Register (TMHSTR)

TMHSTR is a SFR to control to start counting the 16-bit timer n.
This is a write-only register and returns always "0x0000"for reading.

Address : 0xF340 (TMHSTRL/TMHSTR), 0xF341 (TMHSTRH)
Access : W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTR															
Byte	TMHSTRH								TMHSTRL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TH0RUN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is used to start a target timer.
Writing "0": Invalid
Writing "1": Start counting

Bit No.	Bit symbol name	Description (target)
15 to 1	-	Reserved bits
0	TH0RUN	16-bit timer 0

8.2.6 16-Bit Timer Stop Register (TMHSTP)

TMHSTP is a SFR to control to stop counting the 16-bit timer n.
This is a write-only register and returns always "0x0000"for reading.

Address : 0xF342 (TMHSTPL/TMHSTP), 0xF343 (TMHSTPH)
Access : W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTP															
Byte	TMHSTPH								TMHSTPL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TH0STP
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is used to stop a target timer.
Writing "0": Invalid
Writing "1": Stop counting

Bit No.	Bit symbol name	Description (target)
15 to 1	-	Reserved bits
0	TH0STP	16-bit timer 0

8.2.7 16-Bit Timer Status Register (TMHSTAT)

TMHSTAT is a SFR to indicate the status of the 16-bit timer n.

Address : 0xF344 (TMHSTATL/TMHSTAT), 0xF345 (TMHSTATH)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTAT															
Byte	TMHSTATH								TMHSTATL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TH0STAT
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is used to indicate an operating status of a target timer
0: A counting of target timer is stopped (Initial value)
1: A counting of target timer is progress

Bit No.	Bit symbol name	Description (target)
15 to 1	-	Reserved bits
0	TH0STAT	16-bit timer 0

8.3 Description of Operation

8.3.1 Operation Mode

Writing "1" to THnRUN bit causes the 16-bit counter n to start counting up in synchronization with the rising edges of the count clock.

Following two operation modes are available:

- Repeat mode
- One-shot mode

8.3.1.1 Repeat Mode

Figure 8-2 shows the repeat mode operation.

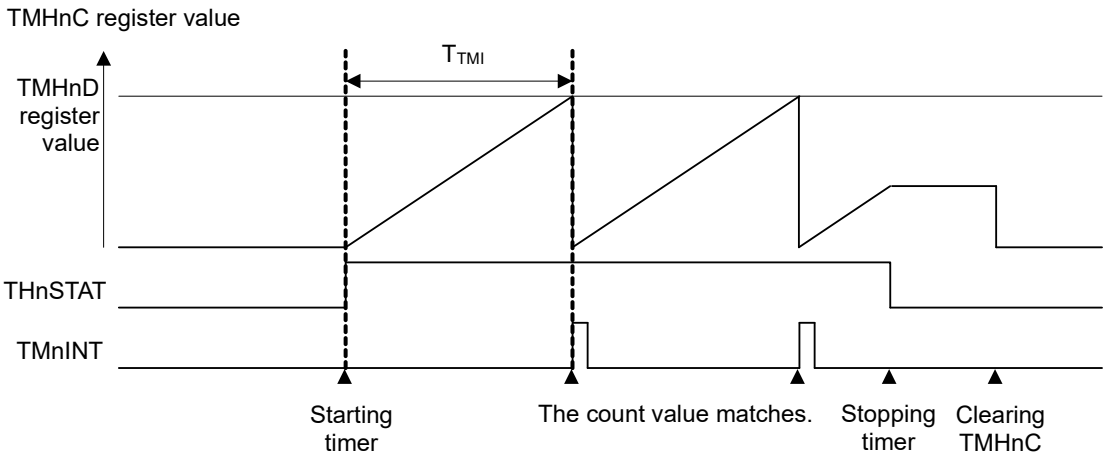


Figure 8-2 Repeat Mode Operation Timing

In the repeat mode, when the timer count value matches with TMHnD register, 16-bit timer n interrupt request (TMnINT) is generated and the output of the port is reversed. Then, the timer count value automatically is reset to "0x0000" and the counting up operation is continued.
TMnINT generation cycle and the port output reverse cycle can be expressed in the following formula:

$$T_{TMI} = \frac{TMHnD + 1}{fTHnCK \quad (Hz)} \qquad (n=0)$$

TMHnD : TMHnD register setting value (0x0001 to 0xFFFF)
fTHnCK : Count clock frequency chosen in TMHnMOD register

See Section 8.3.2 "Start/Stop Timing" for the timing of the timer start/stop and counting up.

8.3.1.2 One-shot Mode

Figure 8-3 shows the one-shot mode operation

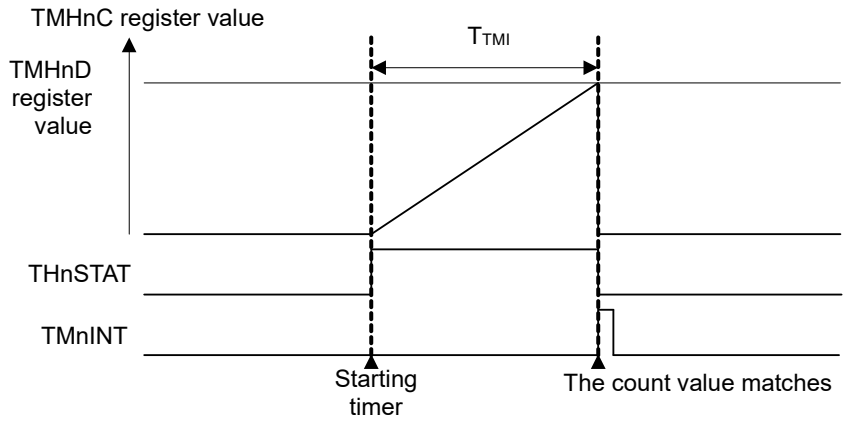


Figure 8-3 One-shot Mode Operation Timing

In the one-shot mode, when the timer count value matches with TMHnD register, 16-bit timer n interrupt (TMnINT) is generated. Then, the timer count value is reset to "0x0000" and the counting is stopped.
TMnINT generation cycle are the same as those in the repeat mode. The same applies to the timer start/stop timing and counting up timing.

8.3.2 Start/Stop Timing

Writing "1" to THnRUN bit of TMHSTR register causes the counting operation to start at the rising edge of the timer clock after the falling edge of the timer clock.
Figure 8-4 shows the timer start timing when the timer clock is LSCLK0 and frequency dividing ratio of the count clock is 1/2 of the timer clock

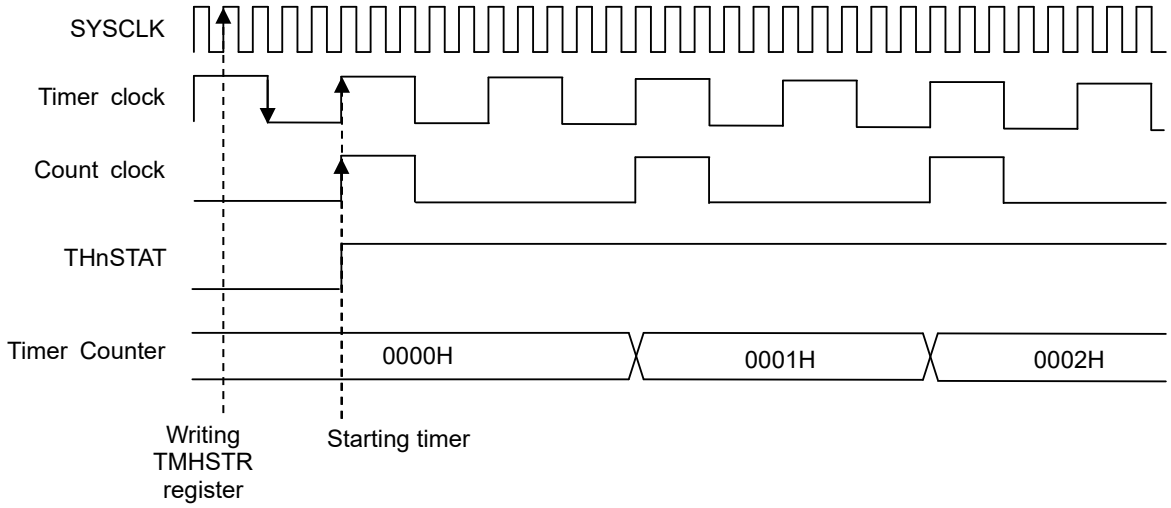


Figure 8-4 Start Timing

Writing "1" to THnSTP bit of TMHSTP register causes the counting operation to stop at the rising edge of the timer clock that follows.
Figure 8-5 shows the timer stop timing when the timer clock is LSCLK0 and frequency dividing ratio of the count clock is 1/2 of the timer clock.

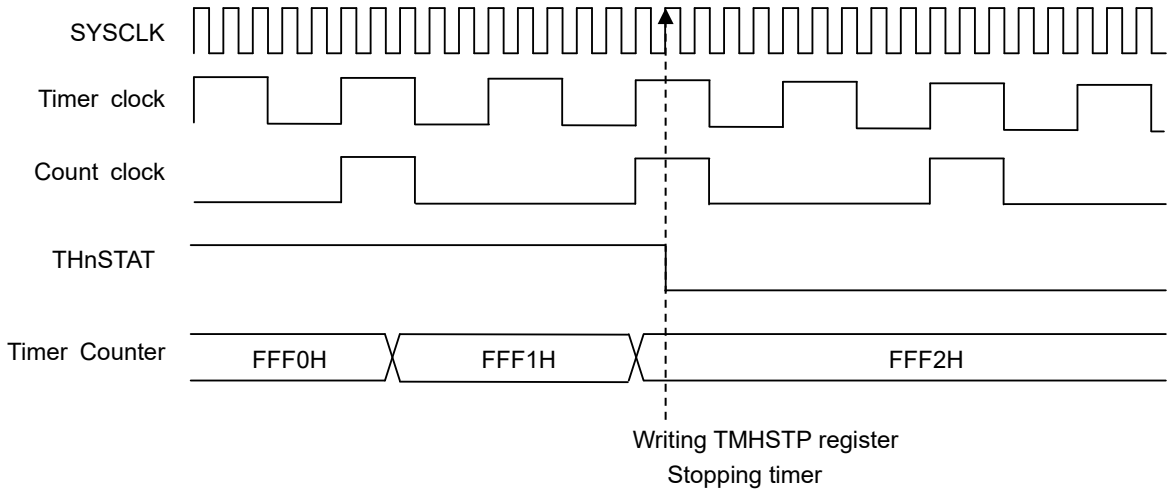


Figure 8-5 Stop Timing

[Note]

- **After THnRUN bit is set to "1", the first interrupt has a time error equivalent to maximum of one clock of the timer clock because the counting operation starts in synchronization with the timer clock. The 2nd timer interrupt or later interrupts have constant cycles.**
- **After THnSTP bit is set to "1", a 16-bit timer n interrupt (TMnINT) may be generated depending on the stop timing because the counting operation stops in synchronization with the timer clock.**

8.3.3 TMHnD Register Update Timing

WhenTMHnD register is rewritten during timer operation, the value written to TMHnD is set at the beginning of the next cycle.
WhenTMHnD register is rewritten while the timer is stopped, the value is set immediately.
Figure 8-6 shows TMHnD register update timing in continuous operation mode.

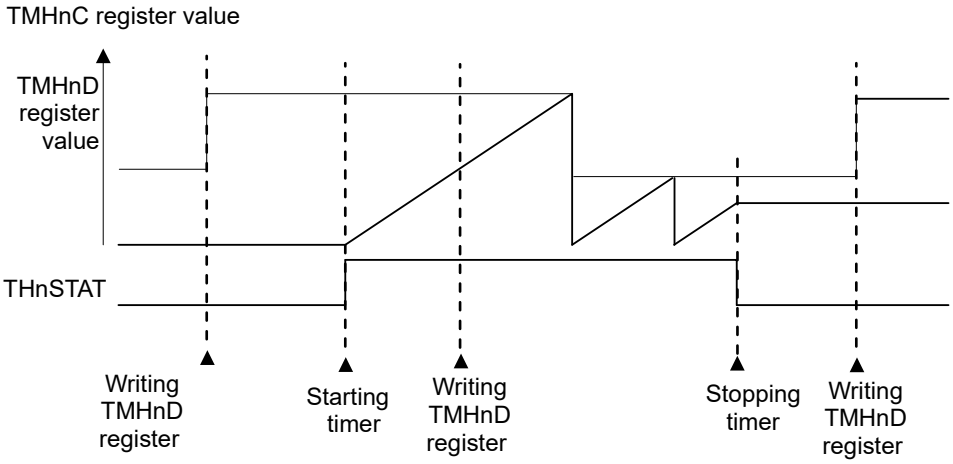


Figure 8-6 Continuous operation mode TMHnD register update timing

Figure 8-7 shows TMHnD register update timing in one-shot operation mode.

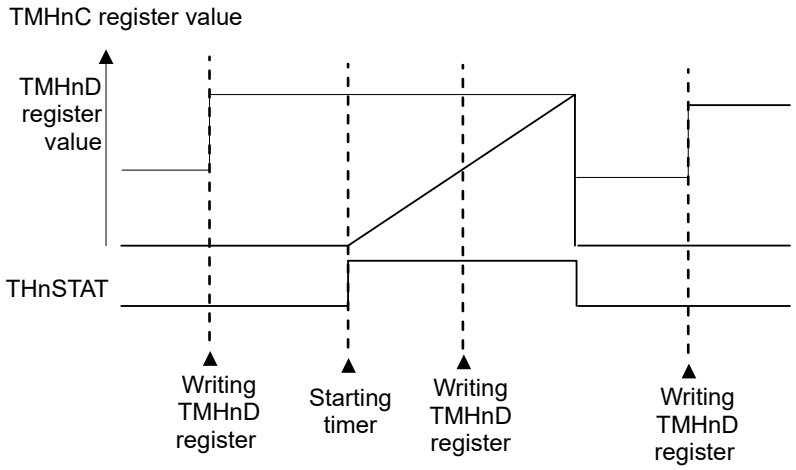


Figure 8-7 one-shot mode TMHnD register update timing

8.3.4 Setting Example

Figure 8-6 shows a setting example.

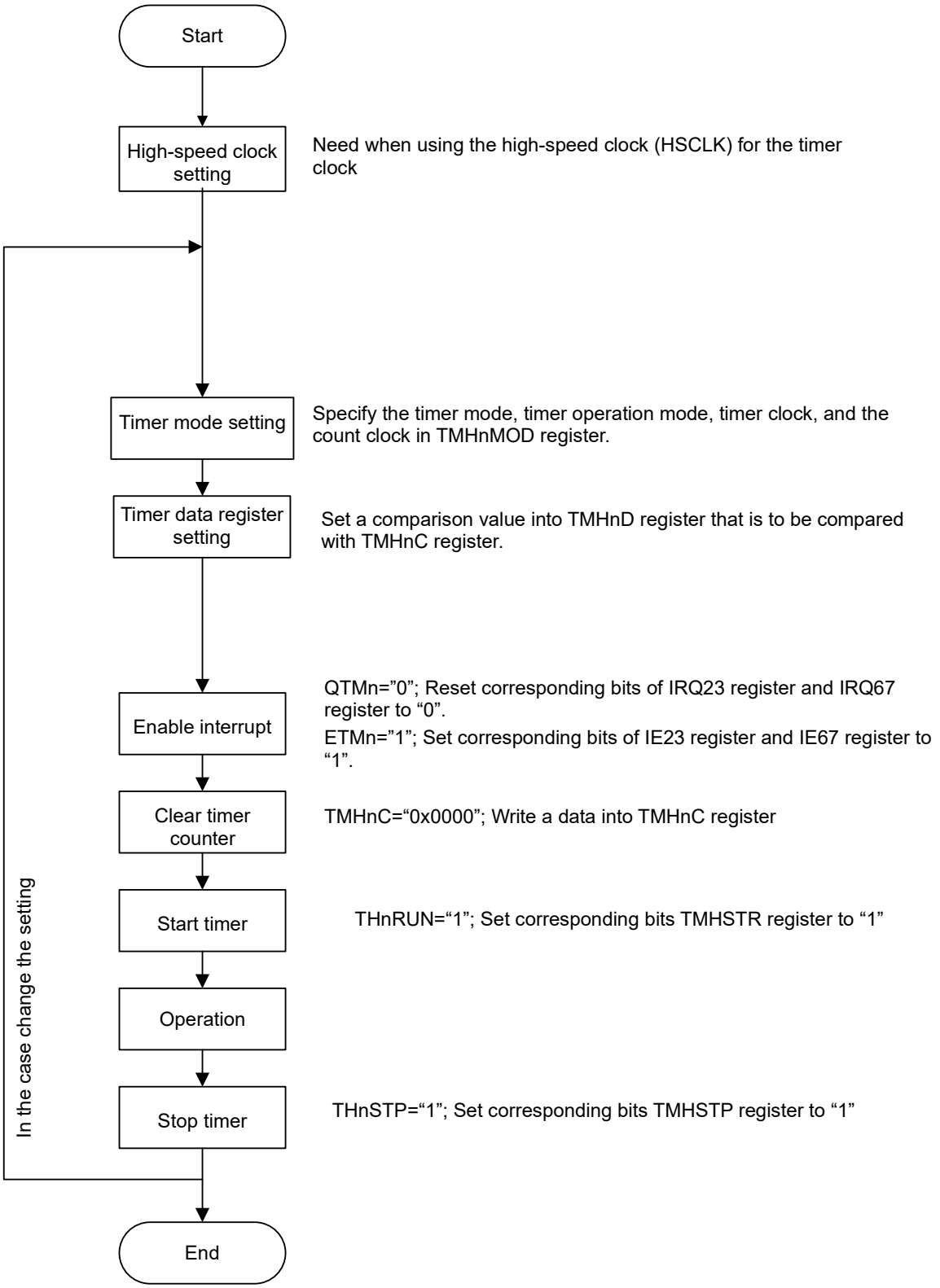


Figure 8-6 Setting Example

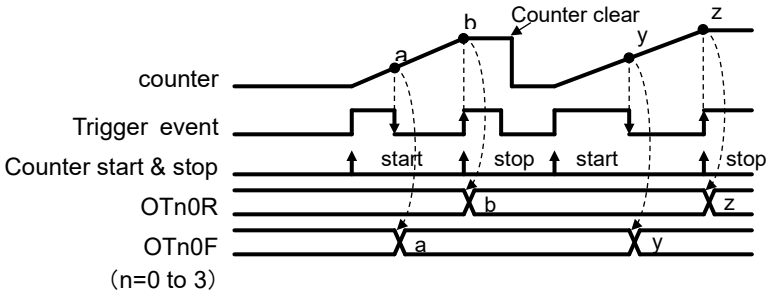
Chapter 9　Operational timer

9. Operational timer

9.1 General Description

Operational timer（OTM）has two operation mode (CAPTURE mode and PWM mode) as described below.

CAPTURE Mode:
In this mode, the counter value is captured.
The captured value at the rising edge of the selected trigger event is stored in OTn0R register,
and the captured value at the falling edge is stored in OTn0F register."



PWM Mode：
This mode outputs a pulse-width-modulated waveform with configurable dead time and duty cycle.
Two types of pulse-width-modulated waveforms (OTMn0 and OTMn1) with different duties are generatable with a common period for each channel.
Additionally, a pulse-width-modulated waveform are synthesizable with a pulse-width-modulated waveform generated by other channels or with an output of the analog comparator.
Refer to table 9-2 for the pin assignment of OTOnA and OTOnB.
The rise timing of the modulated OTMn0 output is set with the setting value of OTn0R register.
The fall timing of the modulated OTMn0 output is set with the setting value of OTn0F register.
And

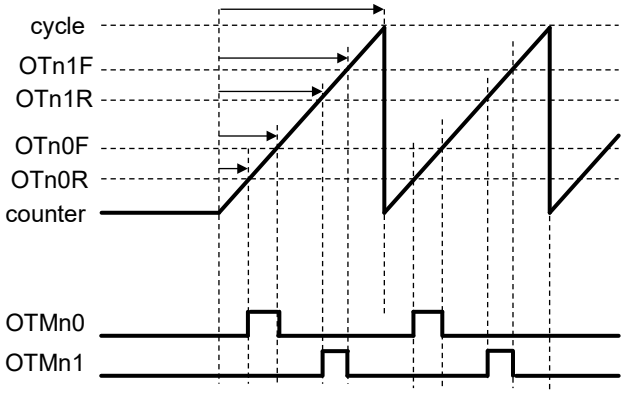


Table 9-1 shows the number of outputs for each channel.

Table 9-1 Number of output of each channel

Channel Number（n）	OTMn0	OTMn1
0	●	●
1	●	●
2	●	-
3	●	-
4	●	●
5	●	●

●：available -：not available

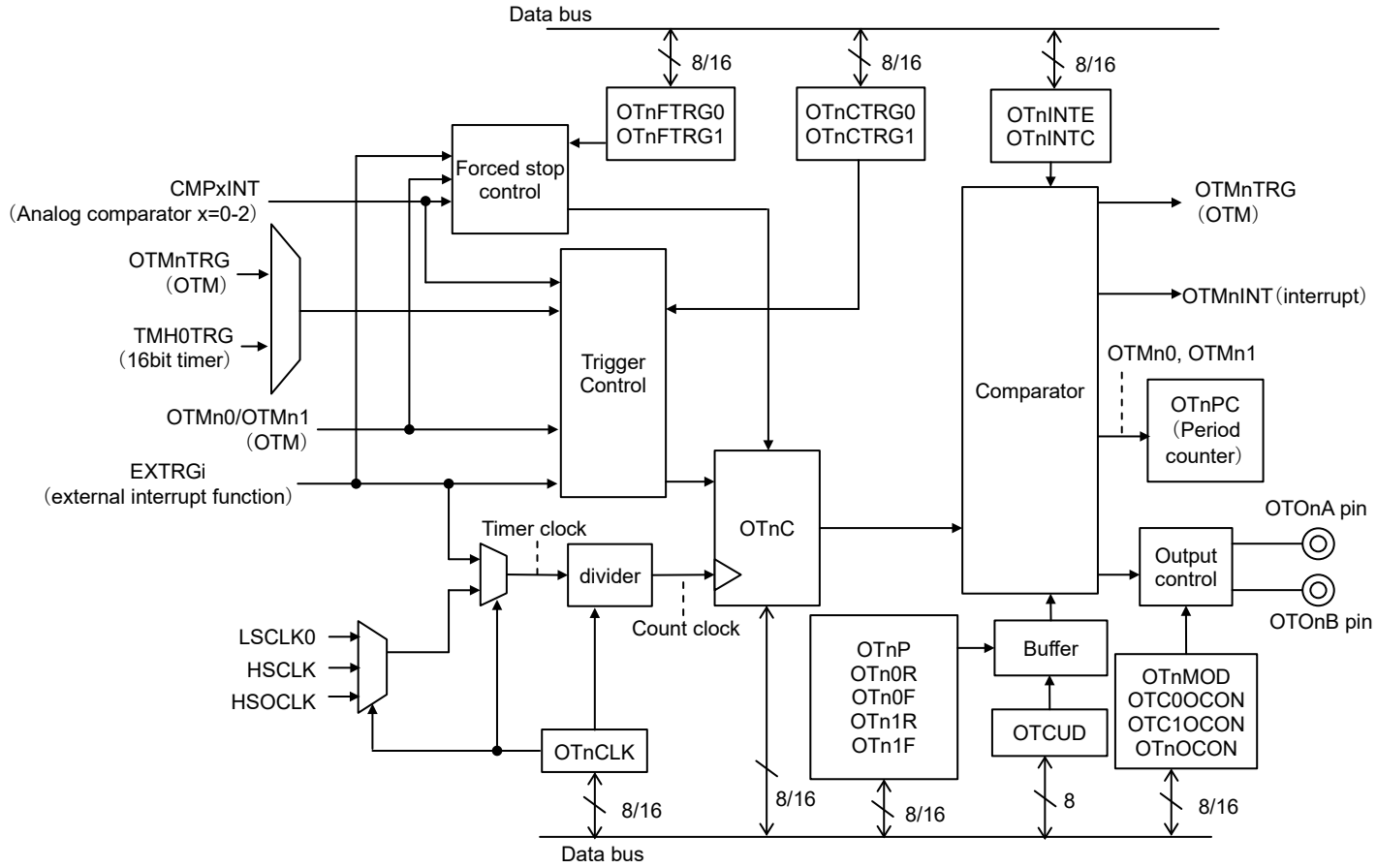
9.1.1 Features

- Capture/PWM functionality using the 16bit counter

- Selectable count clock from LSCLK0（32kHz）, HSCLK（16MHz）, HSOCLK（64MHz）, and from 1 to 128 dividers of an external trigger input.
- Logical switching of a timer output (positive logic/negative logic).
- Logical AND output with another channel's pulse-width-modulated waveform output, an external trigger, and an analog comparator output.
- In addition to periodic interrupts, interrupts that match various setting values such as duty settings is generatable.
- One shot mode.
- Adjustable timing to stop operation in continuous mode.
- Counter control (Start / Stop / Counter clear) by an external trigger input, a Timer interrupt request, an analog comparator output.
- An external trigger input, a timer interrupt request (event trigger), and an analog comparator output enable counter operation start, stop, and clear counters.
- Counter operation can be started/stopped/cleared by an external trigger input, a timer interrupt request (event trigger), and an analog comparator output.
- Forced stop of a terminal output by an external trigger input or an analog comparator output, and an interrupt generated by an external trigger input or an analog comparator output forced stop.
- Output a different duty pulse-width-modulated pulse with same cycle.
- Duty and cycle of an input signal can be measured by the capture function.

9.1.2 Configuration

Figure 9-1 shows the OTM circuit configuration



OTnP	:	OTMn period register
OTn0R	:	OTMn0 Rise Point register
OTn0F	:	OTMn0 Fall Point register
OTn1R	:	OTMn1 Rise Point register
OTn1F	:	OTMn1 Fall Point register
OTnC	:	OTMn counter register
OTnCTRG0	:	OTM0 counter trigger 0 register
OTnCTRG1	:	OTM0 counter trigger1 register
OTnFTRG0	:	OTM0 Forced stop trigger0 register
OTnFTRG1	:	OTM0 Forced stop trigger1 register
OTnMOD	:	OTMn mode register
OTC0OCON	:	OTM common OTMn0 output logic permission register
OTC1OCON	:	OTM common OTMn1 output logic permission register
OTnOCON	:	OTMn output logic control register
OTnINTE	:	OTMn interrupt permission register
OTnINTC	:	OTMn interrupt clear register
OTMnTRG	:	Operational timer trigger(n=0 to 5)
EXTRGi	:	External trigger / External clock (i=0 to 3)
TMH0TRG	:	16bit timer trigger

Fig. 9-1 Configuration of Operational timer

9.1.3 Pin list

Operational timer input/output pins are assigned to general port shared function.

Pin name	I / O	Function
EXI0 to 3	I	External trigger / External clock
OTOnA	O	Operational timer n A output
OTOnB	O	Operational timer n B output

(n=0 to 5)

Table 9-2 shows the general-purpose ports and register settings used by operational timers.

Table 9-2 Ports that use the Operational timer function and register setting

Pin name	Shared port		Setting register	Setting value	Availability	
					20pin Product	24pin product
EXI0	Pins assigned by an external interrupt Refer to the section 18 "External interrupt function"				●	●
EXI1					●	●
EXI2					●	●
EXI3					●	●
OTO0A	P03	4th function	P0MOD3	0011_XXXX*1	●	●
OTO0B	P04		P0MOD4	0011_XXXX*1	●	●
OTO1A	P05		P0MOD5	0011_XXXX*1	●	●
OTO1B	P07		P0MOD7	0011_XXXX*1	-	●
	P12		P1MOD2	0011_XXXX*1	●	●
OTO2A	P06		P0MOD6	0011_XXXX*1	●	●
OTO3A	P10		P1MOD0	0011_XXXX*1	●	●
OTO4A	P11		P1MOD1	0011_XXXX*1	●	●
OTO4B	P02		P0MOD2	0011_XXXX*1	●	●
	P21		P2MOD1	0011_XXXX*1	-	●
OTO5A	P23		P2MOD3	0011_XXXX*1	●	●
OTO5B	P13		P1MOD3	0011_XXXX*1	●	●
	P22		P2MOD2	0011_XXXX*1	-	●

●: Available -: Not available

*1: The setting values for XXXX are as follows

XXXX	Port output status
0010	CMOS output
1010	Nch Open drain output(without pull up)
1111	Nch Open drain output(with pull up)

9.2 Register description

9.2.1 List of register

Writing to register for non-loaded channels is not valid. "0x0000" is always read out when reading.

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF380	OTM common update register	OTCUD	-	W	8	0x00
0xF381	Reserved register	-	-	-	-	-
0xF382	OTM common control register	OTCCONL	OTCCON	R/W	8/16	0x00
0xF383		OTCCONH		R/W	8	0x00
0xF384	OTM common start register	OTCSTRL	OTCSTR	W	8/16	0x00
0xF385		OTCSTRH		W	8	0x00
0xF386	OTM common stop register	OTCSTPL	OTCSTP	W	8/16	0x00
0xF387		OTCSTPH		W	8	0x00
0xF388	OTM common status register	OTCSTATL	OTCSTAT	R	8/16	0x00
0xF389		OTCSTATH		R	8	0x00
0xF38A	OTM common break control register	OTCBRK	-	R/W	8	0x3F
0xF38B	Reserved register	-	-	-	-	-
0xF38C	OTM common OTMn0 output logical authorized register	OTC0OCONL	OTC0OCON	R/W	8/16	0x00
0xF38D		OTC0OCONH		R/W	8	0x00
0xF38E	OTM common OTMn1 output logical authorized register	OTC1OCONL	OTC1OCON	R/W	8/16	0x00
0xF38F		OTC1OCONH		R/W	8	0x00
0xF390	OTM0 output logical authorized register	OT0OCONL	OT0OCON	R/W	8/16	0x00
0xF391		OT0OCONH		R/W	8	0x00
0xF392	OTM1 output logical authorized register	OT1OCONL	OT1OCON	R/W	8/16	0x00
0xF393		OT1OCONH		R/W	8	0x00
0xF394	OTM2 output logical authorized register	OT2OCONL	OT2OCON	R/W	8/16	0x00
0xF395		OT2OCONH		R/W	8	0x00
0xF396	OTM3 output logical authorized register	OT3OCONL	OT3OCON	R/W	8/16	0x00
0xF397		OT3OCONH		R/W	8	0x00
0xF398	OTM4 output logical authorized register	OT4OCONL	OT4OCON	R/W	8/16	0x00
0xF399		OT4OCONH		R/W	8	0x00
0xF39A	OTM5 output logical authorized register	OT5OCONL	OT5OCON	R/W	8/16	0x00
0xF39B		OT5OCONH		R/W	8	0x00
0xF39C	Reserved register	-	-	-	-	-
0xF39D		-		-	-	-
0xF39E	Reserved register	-	-	-	-	-
0xF39F		-		-	-	-
0xF3A0	OTM0 counter trigger 1 register	OT0CTRG1L	OT0CTRG1	R/W	8/16	0x00
0xF3A1		OT0CTRG1H		R/W	8	0x00
0xF3A2	OTM1 counter trigger 1 register	OT1CTRG1L	OT1CTRG1	R/W	8/16	0x00
0xF3A3		OT1CTRG1H		R/W	8	0x00
0xF3A4	OTM2 counter trigger 1 register	OT2CTRG1L	OT2CTRG1	R/W	8/16	0x00
0xF3A5		OT2CTRG1H		R/W	8	0x00
0xF3A6	OTM3 counter trigger 1 register	OT3CTRG1L	OT3CTRG1	R/W	8/16	0x00
0xF3A7		OT3CTRG1H		R/W	8	0x00
0xF3A8	OTM4 counter trigger 1 register	OT4CTRG1L	OT4CTRG1	R/W	8/16	0x00
0xF3A9		OT4CTRG1H		R/W	8	0x00
0xF3AA	OTM5 counter trigger 1 register	OT5CTRG1L	OT5CTRG1	R/W	8/16	0x00
0xF3AB		OT5CTRG1H		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF3AC to 0xF3EE	Reserved registers	-	-	-	-	-
0xF3F0	OTM Cycle setting method selection register	OTCPCL	OTCPC	R/W	8/16	0x00
0xF3F1		OTCPCH		R/W	8	0x00
0xF3F2 to 0xF3FF	Reserved registers	-	-	-	-	-
0xF400	OTM0 cycle register	OT0PL	OT0P	R/W	8/16	0xFF
0xF401		OT0PH		R/W	8	0xFF
0xF402	OTM00 Rise Point register	OT00RL	OT00R	R/W	8/16	0x00
0xF403		OT00RH		R/W	8	0x00
0xF404	OTM00 Fall Point register	OT00FL	OT00F	R/W	8/16	0x00
0xF405		OT00FH		R/W	8	0x00
0xF406	OTM01 Rise Point register	OT01RL	OT01R	R/W	8/16	0x00
0xF407		OT01RH		R/W	8	0x00
0xF408	OTM01 Fall Point register	OT01FL	OT01F	R/W	8/16	0x00
0xF409		OT01FH		R/W	8	0x00
0xF40A	OTM0 counter register	OT0CL	OT0C	R/W	8/16	0x00
0xF40B		OT0CH		R/W	8	0x00
0xF40C	OTM0 status register	OT0STATL	OT0STAT	R	8/16	0x06
0xF40D		OT0STATH		R	8	0x00
0xF40E	OTM0 mode register	OT0MODL	OT0MOD	R/W	8/16	0x00
0xF40F		OT0MODH		R/W	8	0x40
0xF410	OTM0 cycle stop timing register	OT0SCLRL	OT0SCLR	R/W	8/16	0x00
0xF411		OT0SCLRH		R/W	8	0x00
0xF412	OTM0 clock register	OT0CLKL	OT0CLK	R/W	8/16	0x00
0xF413		OT0CLKH		R/W	8	0x00
0xF414	OTM0 counter trigger 0 register	OT0CTRG0L	OT0CTRG0	R/W	8/16	0x00
0xF415		OT0CTRG0H		R/W	8	0x00
0xF416	OTM0 Forced stop trigger 0 register	OT0FTRG0L	OT0FTRG0	R/W	8/16	0x00
0xF417		OT0FTRG0H		R/W	8	0x00
0xF418	OTM0 Forced stop trigger 1 register	OT0FTRG1L	OT0FTRG1	R/W	8/16	0x00
0xF419		OT0FTRG1H		R/W	8	0x00
0xF41A	OTM0 Interrupt enable Register	OT0INTEL	OT0INTE	R/W	8/16	0x00
0xF41B		OT0INTEH		R/W	8	0x00
0xF41C	OTM0 interrupt status register	OT0INTSL	OT0INTS	R	8/16	0x00
0xF41D		OT0INTSH		R	8	0x00
0xF41E	OTM0 interrupt clear register	OT0INTCL	-	W	8	0x00
0xF41F		OT0INTCH		W	8	0x00
0xF420	OTM1 cycle register	OT1PL	OT1P	R/W	8/16	0xFF
0xF421		OT1PH		R/W	8	0xFF
0xF422	OTM10 Rise Point register	OT10RL	OT10R	R/W	8/16	0x00
0xF423		OT10RH		R/W	8	0x00
0xF424	OTM10 Fall Point register	OT10FL	OT10F	R/W	8/16	0x00
0xF425		OT10FH		R/W	8	0x00
0xF426	OTM11 Rise Point register	OT11RL	OT11R	R/W	8/16	0x00
0xF427		OT11RH		R/W	8	0x00
0xF428	OTM11 Fall Point register	OT11FL	OT11F	R/W	8/16	0x00
0xF429		OT11FH		R/W	8	0x00
0xF42A	OTM1 counter register	OT1CL	OT1C	R/W	8/16	0x00
0xF42B		OT1CH		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF42C	OTM1 status register	OT1STATL	OT1STAT	R	8/16	0x06
0xF42D		OT1STATH		R	8	0x00
0xF42E	OTM1 mode register	OT1MODL	OT1MOD	R/W	8/16	0x00
0xF42F		OT1MODH		R/W	8	0x40
0xF430	OTM1 cycle stop timing register	OT1SCLRL	OT1SCLR	R/W	8/16	0x00
0xF431		OT1SCLRH		R/W	8	0x00
0xF432	OTM1 clock register	OT1CLKL	OT1CLK	R/W	8/16	0x00
0xF433		OT1CLKH		R/W	8	0x00
0xF434	OTM1 counter trigger 0 register	OT1CTRG0L	OT1CTRG0	R/W	8/16	0x00
0xF435		OT1CTRG0H		R/W	8	0x00
0xF436	OTM1 Forced stop trigger 0 register	OT1FTRG0L	OT1FTRG0	R/W	8/16	0x00
0xF437		OT1FTRG0H		R/W	8	0x00
0xF438	OTM1 Forced stop trigger 1 register	OT1FTRG1L	OT1FTRG1	R/W	8/16	0x00
0xF439		OT1FTRG1H		R/W	8	0x00
0xF43A	OTM1 interrupt enable register	OT1INTEL	OT1INTE	R/W	8/16	0x00
0xF43B		OT1INTEH		R/W	8	0x00
0xF43C	OTM1 interrupt status register	OT1INTSL	OT1INTS	R	8/16	0x00
0xF43D		OT1INTSH		R	8	0x00
0xF43E	OTM1 interrupt clear register(L/H)	OT1INTCL	-	W	8	0x00
0xF43F		OT1INTCH		W	8	0x00
0xF440	OTM2 cycle register	OT2PL	OT2P	R/W	8/16	0xFF
0xF441		OT2PH		R/W	8	0xFF
0xF442	OTM20 Rise Point register	OT20RL	OT20R	R/W	8/16	0x00
0xF443		OT20RH		R/W	8	0x00
0xF444	OTM20 Fall Point register	OT20FL	OT20F	R/W	8/16	0x00
0xF445		OT20FH		R/W	8	0x00
0xF446	Reserved register	-	-	-	-	-
0xF447		-		-	-	-
0xF448	Reserved register	-	-	-	-	-
0xF449		-		-	-	-
0xF44A	OTM2 counter register	OT2CL	OT2C	R/W	8/16	0x00
0xF44B		OT2CH		R/W	8	0x00
0xF44C	OTM2 status register	OT2STATL	OT2STAT	R	8/16	0x02
0xF44D		OT2STATH		R	8	0x00
0xF44E	OTM2 mode register	OT2MODL	OT2MOD	R/W	8/16	0x00
0xF44F		OT2MODH		R/W	8	0x00
0xF450	OTM2 cycle stop timing register	OT2SCLRL	OT2SCLR	R/W	8/16	0x00
0xF451		OT2SCLRH		R/W	8	0x00
0xF452	OTM2 clock register	OT2CLKL	OT2CLK	R/W	8/16	0x00
0xF453		OT2CLKH		R/W	8	0x00
0xF454	OTM2 counter trigger 0 register	OT2CTRG0L	OT2CTRG0	R/W	8/16	0x00
0xF455		OT2CTRG0H		R/W	8	0x00
0xF456	OTM2 Forced stop trigger 0 register	OT2FTRG0L	OT2FTRG0	R/W	8/16	0x00
0xF457		OT2FTRG0H		R/W	8	0x00
0xF458	OTM2 Forced stop trigger 1 register	OT2FTRG1L	OT2FTRG1	R/W	8/16	0x00
0xF459		OT2FTRG1H		R/W	8	0x00
0xF45A	OTM2 interrupt enable register	OT2INTEL	OT2INTE	R/W	8/16	0x00
0xF45B		OT2INTEH		R/W	8	0x00
0xF45C	OTM2 interrupt status register	OT2INTSL	OT2INTS	R	8/16	0x00
0xF45D		OT2INTSH		R	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF45E	OTM2 interrupt clear register(L/H)	OT2INTCL	-	W	8	0x00
0xF45F		OT2INTCH		W	8	0x00
0xF460	OTM3 cycle register	OT3PL	OT3P	R/W	8/16	0xFF
0xF461		OT3PH		R/W	8	0xFF
0xF462	OTM30 Rise Point register	OT30RL	OT30R	R/W	8/16	0x00
0xF463		OT30RH		R/W	8	0x00
0xF464	OTM30 Fall Point register	OT30FL	OT30F	R/W	8/16	0x00
0xF465		OT30FH		R/W	8	0x00
0xF466	Reserved register	-	-	-	-	-
0xF467		-		-	-	-
0xF468	Reserved register	-	-	-	-	-
0xF469		-		-	-	-
0xF46A	OTM3 counter register	OT3CL	OT3C	R/W	8/16	0x00
0xF46B		OT3CH		R/W	8	0x00
0xF46C	OTM3 status register	OT3STATL	OT3STAT	R	8/16	0x02
0xF46D		OT3STATH		R	8	0x00
0xF46E	OTM3 mode register	OT3MODL	OT3MOD	R/W	8/16	0x00
0xF46F		OT3MODH		R/W	8	0x00
0xF470	OTM3 cycle stop timing register	OT3SCLRL	OT3SCLR	R/W	8/16	0x00
0xF471		OT3SCLRH		R/W	8	0x00
0xF472	OTM3 clock register	OT3CLKL	OT3CLK	R/W	8/16	0x00
0xF473		OT3CLKH		R/W	8	0x00
0xF474	OTM3 counter trigger 0 register	OT3CTRG0L	OT3CTRG0	R/W	8/16	0x00
0xF475		OT3CTRG0H		R/W	8	0x00
0xF476	OTM3 Forced stop trigger 0 register	OT3FTRG0L	OT3FTRG0	R/W	8/16	0x00
0xF477		OT3FTRG0H		R/W	8	0x00
0xF478	OTM3 Forced stop trigger 1 register	OT3FTRG1L	OT3FTRG1	R/W	8/16	0x00
0xF479		OT3FTRG1H		R/W	8	0x00
0xF47A	OTM3 interrupt enable register	OT3INTEL	OT3INTE	R/W	8/16	0x00
0xF47B		OT3INTEH		R/W	8	0x00
0xF47C	OTM3 interrupt status register	OT3INTSL	OT3INTS	R	8/16	0x00
0xF47D		OT3INTSH		R	8	0x00
0xF47E	OTM3 interrupt clear register(L/H)	OT3INTCL	-	W	8	0x00
0xF47F		OT3INTCH		W	8	0x00
0xF480	OTM4 cycle register	OT4PL	OT4P	R/W	8/16	0xFF
0xF481		OT4PH		R/W	8	0xFF
0xF482	OTM40 Rise Point register	OT40RL	OT40R	R/W	8/16	0x00
0xF483		OT40RH		R/W	8	0x00
0xF484	OTM40 Fall Point register	OT40FL	OT40F	R/W	8/16	0x00
0xF485		OT40FH		R/W	8	0x00
0xF486	OTM41 Rise Point register	OT41RL	OT41R	R/W	8/16	0x00
0xF487		OT41RH		R/W	8	0x00
0xF488	OTM41 Fall Point register	OT41FL	OT41F	R/W	8/16	0x00
0xF489		OT41FH		R/W	8	0x00
0xF48A	OTM4 counter register	OT4CL	OT4C	R/W	8/16	0x00
0xF48B		OT4CH		R/W	8	0x00
0xF48C	OTM4 status register	OT4STATL	OT4STAT	R	8/16	0x06
0xF48D		OT4STATH		R	8	0x00
0xF48E	OTM4 mode register	OT4MODL	OT4MOD	R/W	8/16	0x00
0xF48F		OT4MODH		R/W	8	0x40

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF490	OTM4 cycle stop timing register	OT4SCLRL	OT4SCLR	R/W	8/16	0x00
0xF491		OT4SCLRH		R/W	8	0x00
0xF492	OTM4 clock register	OT4CLKL	OT4CLK	R/W	8/16	0x00
0xF493		OT4CLKH		R/W	8	0x00
0xF494	OTM4 counter trigger 0 register	OT4CTRG0L	OT4CTRG0	R/W	8/16	0x00
0xF495		OT4CTRG0H		R/W	8	0x00
0xF496	OTM4 Forced stop trigger 0 register	OT4FTRG0L	OT4FTRG0	R/W	8/16	0x00
0xF497		OT4FTRG0H		R/W	8	0x00
0xF498	OTM4 Forced stop trigger 1 register	OT4FTRG1L	OT4FTRG1	R/W	8/16	0x00
0xF499		OT4FTRG1H		R/W	8	0x00
0xF49A	OTM4 interrupt enable register	OT4INTEL	OT4INTE	R/W	8/16	0x00
0xF49B		OT4INTEH		R/W	8	0x00
0xF49C	OTM4 interrupt status register	OT4INTSL	OT4INTS	R	8/16	0x00
0xF49D		OT4INTSH		R	8	0x00
0xF49E	OTM4 interrupt clear register(L/H)	OT4INTCL	-	W	8	0x00
0xF49F		OT4INTCH		W	8	0x00
0xF4A0	OTM5 cycle register	OT5PL	OT5P	R/W	8/16	0xFF
0xF4A1		OT5PH		R/W	8	0xFF
0xF4A2	OTM50 Rise Point register	OT50RL	OT50R	R/W	8/16	0x00
0xF4A3		OT50RH		R/W	8	0x00
0xF4A4	OTM50 Fall Point register	OT50FL	OT50F	R/W	8/16	0x00
0xF4A5		OT50FH		R/W	8	0x00
0xF4A6	OTM51 Rise Point register	OT51RL	OT51R	R/W	8/16	0x00
0xF4A7		OT51RH		R/W	8	0x00
0xF4A8	OTM51 Fall Point register	OT51FL	OT51F	R/W	8/16	0x00
0xF4A9		OT51FH		R/W	8	0x00
0xF4AA	OTM5 counter register	OT5CL	OT5C	R/W	8/16	0x00
0xF4AB		OT5CH		R/W	8	0x00
0xF4AC	OTM5 status register	OT5STATL	OT5STAT	R	8/16	0x06
0xF4AD		OT5STATH		R	8	0x00
0xF4AE	OTM5 mode register	OT5MODL	OT5MOD	R/W	8/16	0x00
0xF4AF		OT5MODH		R/W	8	0x40
0xF4B0	OTM5 cycle stop timing register	OT5SCLRL	OT5SCLR	R/W	8/16	0x00
0xF4B1		OT5SCLRH		R/W	8	0x00
0xF4B2	OTM5 clock register	OT5CLKL	OT5CLK	R/W	8/16	0x00
0xF4B3		OT5CLKH		R/W	8	0x00
0xF4B4	OTM5 counter trigger 0 register	OT5CTRG0L	OT5CTRG0	R/W	8/16	0x00
0xF4B5		OT5CTRG0H		R/W	8	0x00
0xF4B6	OTM5 Forced stop trigger 0 register	OT5FTRG0L	OT5FTRG0	R/W	8/16	0x00
0xF4B7		OT5FTRG0H		R/W	8	0x00
0xF4B8	OTM5 Forced stop trigger 1 register	OT5FTRG1L	OT5FTRG1	R/W	8/16	0x00
0xF4B9		OT5FTRG1H		R/W	8	0x00
0xF4BA	OTM5 interrupt enable register	OT5INTEL	OT5INTE	R/W	8/16	0x00
0xF4BB		OT5INTEH		R/W	8	0x00
0xF4BC	OTM5 interrupt status register	OT5INTSL	OT5INTS	R	8/16	0x00
0xF4BD		OT5INTSH		R	8	0x00
0xF4BE	OTM5 interrupt clear register(L/H)	OT5INTCL	-	W	8	0x00
0xF4BF		OT5INTCH		W	8	0x00

9.2.2 OTMn cycle register OTnP：n=0 to 5）

OTnP is a SFR to set OTMn cycle (clock count)
The configurable range is 0x0001 to 0xFFFF (clock count:2 to 65536)

If "1" is written to either OTnPC0 or OTnPC1 bits of OTCPC register, the read value of OTnP register is the buffer value set in OTCPC register..
See "9.3.7 Changes during periodic and duty operation" for details on buffers. Writing to OTnP register is invalid in this time.

Address： 0xF400（OT0PL/OT0P），0xF401（OT0PH），0xF420（OT1PL/OT1P），0xF421（OT1PH），
0xF440（OT2PL/OT2P），0xF441（OT2PH），0xF460（OT3PL/OT3P），0xF461（OT3PH），
0xF480（OT4PL/OT4P），0xF481（OT4PH），0xF4A0（OT5PL/OT5P），0xF4A1（OT5PH）
Access： R/W
Access size：8bit/16bit
Initial value：0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnP															
Byte	OTnPH								OTnPL							
Bit	OTnP15	OTnP14	OTnP13	OTnP12	OTnP11	OTnP10	OTnP9	OTnP8	OTnP7	OTnP6	OTnP5	OTnP4	OTnP3	OTnP2	OTnP1	OTnP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 0	OTnP15 to OTnP0	0x0001 to 0xFFFF: Set the period to the set value in OTnP register + 1 clock.

【Note】

- **When 0x0000 is written in this register, 0x0001 is set and the read value is also** becoming **0x0001.**

9.2.3 OTMn0 Rise Point register（OTn0R：n=0 to 5）

OTn0R is a SFR to set rise timing of output or store a captured data.
In CAPTURE mode, the register is read only (writing is invalid).

Address: 0xF402(OT00RL/OT00R), 0xF403(OT00RH), 0xF422(OT10RL/OT10R), 0xF423(OT10RH),
0xF442(OT20RL/OT20R), 0xF443(OT20RH), 0xF462(OT30RL/OT30R), 0xF463(OT30RH),
0xF482(OT40RL/OT40R), 0xF483(OT40RH), 0xF4A2(OT50RL/OT50R), 0xF4A3(OT50RH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTn0R															
Byte	OTn0RH								OTn0RL							
Bit	OTn0R 15	OTn0R 14	OTn0R 13	OTn0R 12	OTn0R 11	OTn0R 10	OTn0R 9	OTn0R 8	OTn0R 7	OTn0R 6	OTn0R 5	OTn0R 4	OTn0R 3	OTn0R 2	OTn0R 1	OTn0R 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	OTn0R15 to OTn0R0	CAPTURE mode 0x0000 to 0xFFFF: Captured count value is stored. Reading out of this register clears the status bit which indicate capture status shown below. - OTnFLGA bit of OTnSTAT register - OTnIS0R bit of OTnINTS register PWM mode 0x0000 to 0xFFFF: Set rise timing of OTMn0 output. OTMn0 output becomes "H" when counter value = OTn0R. If same value is set to both OTn0R and OTn0F, the duty becomes 0%.

9.2.4 OTMn0 Fall Point register（OTn0F：n=0 to 5）

OTn0F is a SFR to set Fall timing of output or store a captured data.
In CAPTURE mode, the register is read only (writing is invalid).

Address: 0xF404(OT00FL/OT00F), 0xF405(OT00FH), 0xF424(OT10FL/OT10F), 0xF425(OT10FH),
0xF444(OT20FL/OT20F), 0xF445(OT20FH), 0xF464(OT30FL/OT30F), 0xF465(OT30FH),
0xF484(OT40FL/OT40F), 0xF485(OT40FH), 0xF4A4(OT50FL/OT50F), 0xF4A5(OT50FH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTn0F															
Byte	OTn0FH								OTn0FL							
Bit	OTn0F 15	OTn0F 14	OTn0F 13	OTn0F 12	OTn0F 11	OTn0F 10	OTn0F 9	OTn0F 8	OTn0F 7	OTn0F 6	OTn0F 5	OTn0F 4	OTn0F 3	OTn0F 2	OTn0F 1	OTn0F 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	OTn0F15 to OTn0F0	CAPTURE mode 0x0000 to 0xFFFF: Captured count value is stored. Reading out of this register clears status bit which indicate capture status shown below. - OTnFLGB bit of OTnSTAT register - OTnIS0F bit of OTnINTS register PWM mode 0x0000 to 0xFFFF: Set fall timing of OTMn0 output. OTMn0 output becomes "L" when counter value = OTn0F. If same value is set to both OTn0R and OTn0F, the duty becomes 0%.

9.2.5 OTMn1 Rise Point register（OTn1R：n=0, 1, 4, 5）

OTn1R is a SFR to set rising time of OTMn1 output.

Address: 0xF406（OT01RL/OT01R）, 0xF407（OT01RH）, 0xF426（OT11RL/OT11R）, 0xF427（OT11RH）,
0xF486（OT41RL/OT41R）, 0xF487（OT41RH）, 0xF4A6（OT51RL/OT51R）, 0xF4A7（OT51RH）
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTn1R															
Byte	OTn1RH								OTn1RL							
Bit	OTn1R 15	OTn1R 14	OTn1R 13	OTn1R 12	OTn1R 11	OTn1R 10	OTn1R 9	OTn1R 8	OTn1R 7	OTn1R 6	OTn1R 5	OTn1R 4	OTn1R 3	OTn1R 2	OTn1R 1	OTn1R 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	OTn1R15 to OTn1R0	・CAPTURE mode Unused. The reading value becomes "0x0000". ・PWM mode 0x0000 to 0xFFFF:Set OTMn1 output rise timing. OTMn1 output becomes "H" when counter value = OTn1R. If same value is set to both OTn1R and OTn1F, the duty becomes 0%.

9.2.6 OTMn1 Fall Point register（OTn1F：n=0，1，4，5）

OTn1F is a SFR to set falling time of OTMn1 output.

Address: 0xF408(OT01FL/OT01F), 0xF409(OT01FH), 0xF428(OT11FL/OT11F), 0xF429(OT11FH),
0xF488(OT41FL/OT41F), 0xF489(OT41FH), 0xF4A8(OT51FL/OT51F), 0xF4A9(OT51FH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTn1F															
Byte	OTn1FH								OTn1FL							
bit	OTn1F 15	OTn1F 14	OTn1F 13	OTn1F 12	OTn1F 11	OTn1F 10	OTn1F 9	OTn1F 8	OTn1F 7	OTn1F 6	OTn1F 5	OTn1F 4	OTn1F 3	OTn1F 2	OTn1F 1	OTn1F 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	OTn1F15 to OTn1F0	・CAPTURE mode Unused. The reading value becomes "0x0000". ・PWM mode 0x0000 to 0xFFFF: Set OTMn1 output fall timing. OTMn1 output becomes "L" when counter value = OTn1F. If same value is set to both OTn1R and OTn1F, the duty becomes 0%.

9.2.7 OTMn counter register（OTnC：n=0 to 5）

OTnC is a SFR to monitor OTMn count value.
When any data is written, OTnC is cleared to "0x0000".

Address: 0xF40A（OT0CL/OT0C），0xF40B（OT0CH），0xF42A（OT1CL/OT1C），0xF42B（OT1CH），
0xF44A（OT2CL/OT2C），0xF44B（OT2CH），0xF46A（OT3CL/OT3C），0xF46B（OT3CH），
0xF48A（OT4CL/OT4C），0xF48B（OT4CH），0xF4AA（OT5CL/OT5C），0xF4AB（OT5CH）
Access: R/W
Access size: 8/16bit
initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnC															
Byte	OTnCH								OTnCL							
Bit	OTnC15	OTnC14	OTnC13	OTnC12	OTnC11	OTnC10	OTnC9	OTnC8	OTnC7	OTnC6	OTnC5	OTnC4	OTnC3	OTnC2	OTnC1	OTnC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]
If the timer clock frequency is same as the system clock frequency, a proper counted value is readout.
If the timer clock frequency is faster than the system clock frequency, a wrong value is readout.
If the timer clock frequency is slower than the system clock frequency, read OTnC multiple times and determine the validity from the period difference between the timer clock and the system clock and the increase in the count value.

9.2.8 OTMn status register（OTnSTAT：n=0 to 5）

OTnSTAT is a SFR to indicate OTMn status.

Address: 0xF40C（OT0STATL/OT0STAT）, 0xF40D（OT0STATH）,
0xF42C（OT1STATL/OT1STAT）, 0xF42D（OT1STATH）,
0xF44C（OT2STATL/OT2STAT）, 0xF44D（OT2STATH）,
0xF46C（OT3STATL/OT3STAT）, 0xF46D（OT3STATH）,
0xF48C（OT4STATL/OT4STAT）, 0xF48D（OT4STATH）,
0xF4AC（OT5STATL/OT5STAT）, 0xF4AD（OT5STATH）
Access: R
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnSTAT															
Byte	OTnSTATH								OTnSTATL							
Bit	-	-	-	-	-	-	-	-	OTnSTA	OTnFLGC	OTnFLGB	OTnFLGA	-	OTnOBSTA*	OTnOASTA	OTnUD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bit
7	OTnSTA	Indicate OTMn operating status. 0: Counter stopped (initial value) 1: Counter in operation
6	OTnFLGC	Indicate whether the setting of OTnCST bit of OTnMOD register is valid or not. When OTnFLGC bit is set to "1", the counter start by the counter control trigger is disabled. When the value of OTnC register is read out, it is automatically cleared. 0: Startable status by event trigger (initial value) 1: Un-startable status by event trigger
5	OTnFLGB	Indicate flag B status. ・CAPTURE mode 0: No capture data (initial value) 1: With capture data (cleared by reading out OTn0F register) ・PWM mode*1 0: "L" is output from OTOnB pin. (Initial value) 1: "H" is output from OTOnB pin. *1) CH2, CH3 (n=2,3) is always fixed at 0 in PWM mode.
4	OTnFLGA	Indicate flag A status. ・CAPTURE mode 0: No capture data (initial value) 1: With capture data (cleared by reading out OTn0R register) ・PWM mode 0: "L" is output from OTOnA pin. (Initial value) 1: "H" is output from OTOnA pin.
3	-	Reserved bit
2	OTnOBSTA	Indicate OTMn1 output status *Not available in CH2 and CH3. ・CAPTURE mode The read value of this bit is always fixed at 0. ・PWM mode*1 0: Indicate that OTMn1 output is as set (initial value). 1: Indicate that OTMn1 output is fixed to the value set by OTnSTPON bit of OTnMOD register. *1) CH2, CH3 (n=2, 3) are always fixed to 0 in PWM mode.

Bit No.	Bit symbol name	Description
1	OTnOASTA	Indicate OTMn0 output status. ・CAPTURE mode The read value of this bit is always fixed at 0. ・PWM mode 0: Indicate that OTMn0 output is as set (initial value). 1: Indicate that OTMn0 output is fixed to the value set by OTnSTPOP bit of OTnMOD register.
0	OTnUD	Indicate the status of the completion after generating an update request of OTnP register or OTn0R/OTn0F/OTn1R/OTn1F register by writing "1" to OTCUDn bit of OTCUD register. When the transfer is completed, this bit is cleared automatically. 0: The update is completed (Initial value) 1: Requesting the update

[Note]
When switching from PWM mode to CAPTURE mode, OTnFLGA bit and OTnFLGB bits might be unintentionally set to "1".
So, when switching to CAPTURE mode, read OTn0R register and OTn0F register once to initialize OTnFLGA and OTnFLGB bits.

9.2.9 OTMn mode register（OTnMOD：n=0 to 5）

OTnMOD is a SFR to select OTMn output.

Address: 0xF40E（OT0MODL/OT0MOD），0xF40F（OT0MODH），
0xF42E（OT1MODL/OT1MOD），0xF42F（OT1MODH），
0xF44E（OT2MODL/OT2MOD），0xF44F（OT2MODH），
0xF46E（OT3MODL/OT3MOD），0xF46F（OT3MODH）
0xF48E（OT4MODL/OT4MOD），0xF48F（OT4MODH），
0xF4AE（OT5MODL/OT5MOD），0xF4AF（OT5MODH）

Access: R/W

Access size: 8/16bit

Initial value: 0x4000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnMOD															
Byte	OTnMODH								OTnMODL							
Bit	OTnOSL1*	OTnOSL0*	OTnOSNB*	OTnOSNA	-	-	OTnSTPON*	OTnSTPOP	OTnOST	OTnSTSYN	OTnDCLH	OTnCST	-	-	-	OTnMD0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Bit No.	Bit symbol name	Description
15	OTnOSL1	Selects the phase of the signal to be output from OTOnB pin. *Not available in CH2 and CH3. 0: Output OTMn1 from OTOnB pin (Initial value) 1: Output OTMn0 from OTOnB pin
14	OTnOSL0	Selects the phase of the signal to be output from OTOnA pin. *Not available in CH2 and CH3. OTO1A and OTO3A is fixed to 0: Output OTMn1 from OTOnA pin (Initial value) 1: Output OTMn0 from OTOnA pin
13	OTnOSNB	Inverts OTOnB pin output signal. *Not available in CH2 and CH3. Invert the signal selected by OTnOSL1 (bit 15). 0: No invert the output (initial values) 1: Invert the output
12	OTnOSNA	Inverts OTOnA pin output signal. Invert the signal selected by OTnOSL0 (bit 14). 0: No invert the output (initial values) 1: Invert the output
11, 10	-	Reserved bit
9	OTnSTPON	Set OTMn1 output state during the counter is stopped or during the output is stopped in forced stop state. *Not available in CH2 and CH3. ・CAPTURE mode Not used. ・PWM mode 0: Set OTMn1 output "L" when stopped. (Initial value) If the counter is restarted without clearing, the state will be "L" until the next cycle. 1: Set OTMn1 output "H" when stopped. If the counter is restarted without clearing, the state will be "H" until the next cycle.
8	OTnSTPOP	Set OTMn0 output state during the counter is stopped or during the output is stopped in forced stop state. ・CAPTURE mode Not used. ・PWM mode 0: Set OTMn0 output "L" when stopped. (Initial value) If the counter is restarted without clearing, the state will be "L" until the next cycle. 1: Set OTMn0 output "H" when stopped. If the counter is restarted without clearing, the state will be "H" until the next cycle.

Bit No.	Bit symbol name	Description
7	OTnOST	Set OTMn operation mode (continuous mode/one-shot mode). ・CAPTURE mode 0: Auto mode (Initial value) Whenever a capture trigger comes to OTn0R and OTn0F registers, these registers are updated to the captured data. 1: Single mode Once the data is captured to OTn0R or OTn0F register, the next capture will not be performed until it is readout. ・PWM mode 0: Continuous mode (initial value) 1: One-shot mode
6	OTnSTSYN	Set OTMn stop timing 0: Stop at the timing of writing the corresponding bit of OTCSTP register or stop when a stop input is received due to a trigger event. (Initial value) 1: Stop after writing the corresponding bit of OTCSPT register, or after accepting a stop input by a trigger event, wait for the interval set by OTnSCLR register from the end of the period and stop. The setting of this bit is enabled when the software is stopped or when it is stopped by a trigger event. It is not valid for stopping by force stop. In capture mode and one-shot mode, the setting of this bit is disabled. When this bit is set to "1", if a software stop or trigger event stop is entered, and a count start by software or trigger input is entered before the end of the cycle is reached, the counting operation will continue without stopping at the end of the cycle.
5	OTnDCLH	Disable counter clear by a trigger event when OTMn0 level is "H". ・CAPTURE mode Unused ・PWM mode 0: Clear enabled regardless of OTMn0 level (initial value) 1: Clear disabled when OTMn0 level is "H"
4	OTnCST	Select a counter start operation mode by trigger event. 0: During the counter is stopped (excluding forced stop), the counter is always started by the trigger event. (Initial value) 1: During counter stop (excluding forced stop), the trigger event is used until OTnC register is read. The counter does not start.
3 to 1	-	Reserved bit
0	OTnMD0	Set OTMn mode 0: PWM mode (initial value) 1: CAPTURE mode

[Note]
Set OTnMOD register when the counter operation is stopped.

9.2.10 OTMn Output logic control register（OTnOCON：n=0 to 5）

OTnOCON is a SFR to set OTMn pin output function.
This register is enabled when the logical output permission state is set with OTM common OTMn0/1 output logical authorized register (OTC0/1OCON).

Address: 0xF390（OT0OCONL/OT0OCON），0xF391（OT0OCONH），
0xF392（OT1OCONL/OT1OCON），0xF393（OT1OCONH），
0xF394（OT2OCONL/OT0OCON），0xF395（OT2OCONH），
0xF396（OT3OCONL/OT1OCON），0xF397（OT3OCONH）
0xF398（OT4OCONL/OT4OCON），0xF399（OT4OCONH），
0xF39A（OT5OCONL/OT5OCON），0xF39B（OT5OCONH）
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnOCON															
Byte	OTnOCONH								OTnOCONL							
Bit	OTnAND17*	OTnAND16*	OTnAND15*	OTnAND14*	OTnAND13*	OTnAND12*	OTnAND11*	OTnAND10*	OTnAND07	OTnAND06	OTnAND05	OTnAND04	OTnAND03	OTnAND02	OTnAND01	OTnAND00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Bit No.	Bit symbol name	Description
15 to12	OTnAND17 to OTnAND14	Select an output signal to take Logical AND with OTMn1 output. It is enabled when OTn1OEN1 bit of OTC1OCON register is "1". See Table 9-3A for signals that can be Logical AND.
11 to 8	OTnAND13 to OTnAND10	Select an output signal to take Logical AND with OTMn1 output. It is enabled when OTn1OEN0 bit of OTC1OCON register is "1". See Table 9-3B for signals that can be Logical AND.
7 to 4	OTnAND07 to OTnAND04	Select an output signal to take Logical AND with OTMn0 output. It is enabled when OTn0OEN1 bit of OTC0OCON register is "1". See Table 9-3A for signals that can be Logical AND.
3 to 0	OTnAND03 to OTnAND00	Select an output signal to take Logical AND with OTMn0 output. It is enabled when OTn0OEN0 bit of OTC0OCON register is "1". See Table 9-3B for signals that can be Logical AND.

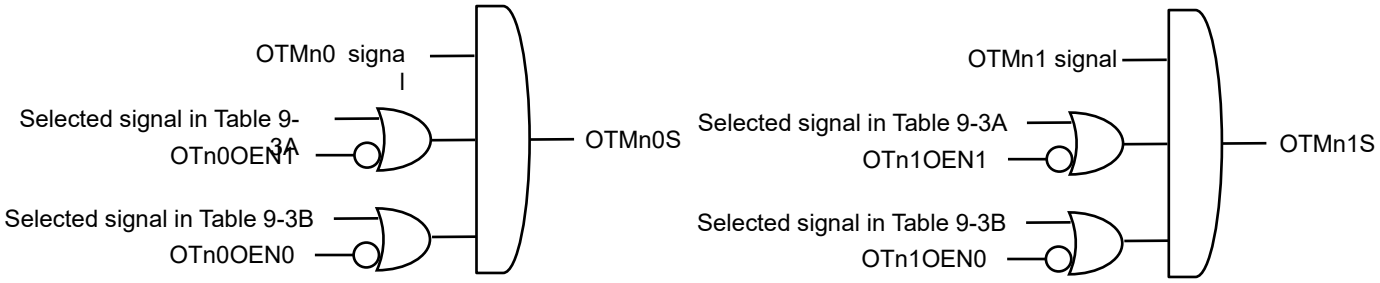


Fig. 9-3 OTnOCON register Configuration Diagram

Table 9-3A OTnAND17 to 14,OTnAND07 to 04 Correspondence table

OTnAND17 to 14 OTnAND07 to 04	OTMn0/OTMn1
0000	OTM30S
0001	OTM40S
0010	OTM41S
0011	OTM50S
0100	OTM51S
0101	Inverted of OTM30S
0110	Inverted of OTM40S
0111	Inverted of OTM41S
1000	Inverted of OTM50S
1001	Inverted of OTM51S
1010	Analog comparator CH0
1011	Analog comparator CH2
1100	Inverted of Analog comparator CH0
1101	Inverted of Analog comparator CH2
1110	External trigger 2 input(EXTRG2)
1111	External trigger 3 input(EXTRG3)

Table 9-3B OTnAND13 to 10,OTnAND03 to 00 Correspondence table

OTnAND13 to 10 OTnAND03 to 00	OTMn0/OTMn1
0000	OTM00S
0001	OTM01S
0010	OTM10S
0011	OTM11S
0100	OTM20S
0101	Inverted of OTM00S
0110	Inverted of OTM01S
0111	Inverted of OTM10S
1000	Inverted of OTM11S
1001	Inverted of OTM20S
1010	Analog comparator CH0
1011	Analog comparator CH1
1100	Inverted of Analog comparator CH0
1101	Inverted of Analog comparator CH1
1110	External trigger 0 input(EXTRG0)
1111	External trigger 1 input(EXTRG1)

[Note]
If it is required to update OTnOCON register while the counter is running, change the setting value after stopping the logical output by OTC0OCON/OTC1OCON register.
If the setting of OTnOCON register is changed during counter operation in the logical output permitting state, unintended waveforms might be output.

9.2.11 OTMn cycle stop timing register（OTnSCLR：n=0 to 5）

OTnSCLR is a SFR to set a counter stop timing at OTMn cycle end.
Setting of the SFR is disabled when operating in CAPTURE mode and One-shot mode.

Address: 0xF410（OT0SCLRL/OT0SCLR），0xF411（OT0SCLRH），
0xF430（OT1SCLRL/OT1SCLR），0xF431（OT1SCLRH），
0xF450（OT2SCLRL/OT2SCLR），0xF451（OT2SCLRH），
0xF470（OT3SCLRL/OT3SCLR），0xF471（OT3SCLRH），
0xF490（OT4SCLRL/OT4SCLR），0xF491（OT4SCLRH），
0xF4B0（OT5SCLRL/OT5SCLR），0xF4B1（OT5SCLRH）

Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnSCLR															
Byte	OTnSCLRH								OTnSCLRL							
Bit	-	-	-	-	-	-	-	-	OTnDLY7	OTnDLY6	OTnDLY5	OTnDLY4	OTnDLY3	OTnDLY2	OTnDLY1	OTnDLY0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bit
7 to 0	OTnDLY7 to OTnDLY0	When stop the counter at the end of the cycle by setting OTnSTSYN bit of OTnMOD register to "1", the counter stops after the time interval set by this bit has elapsed. ・PWM mode 0x00 to 0xFF: Set OTMn counter stop time interval. Time interval is calculated by the below formula. Time interval (between when the counter is stopped after the count is completed) = count clock period x OTnDLY setting value

[Note]
- **Set OTnSCLR register when counter operation is stopped.**
- **The time interval set to OTnDLY bit must be shorter than the interval set to OTnP register. If the time interval is set for OTnDLY bit longer than the interval set to OTnP register, the counter may stop even if the time interval is less than the interval set by OTnDLY bit.**

9.2.12 OTMn clock register（OTnCLK：n=0 to 5）

OTnCLK is a SFR to set timer clock of OTMn.

Address: 0xF412（OT0CLKL/OT0CLK），0xF413（OT0CLKH），0xF432（OT1CLKL/OT1CLK），0xF433（OT1CLKH），
0xF452（OT2CLKL/OT2CLK），0xF453（OT2CLKH），0xF472（OT3CLKL/OT3CLK），0xF473（OT3CLKH），
0xF492（OT4CLKL/OT4CLK），0xF493（OT4CLKH），0xF4B2（OT5CLKL/OT5CLK），0xF4B3（OT5CLKH）
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnCLK															
Byte	OTnCLKH								OTnCLKL							
Bit	-	-	-	-	-	-	-	-	-	-	OTnCKD1	OTnCKD0	-	OTnCK2	OTnCK1	OTnCK0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 6	-	Reserved bit
5 to 4	OTnCKD1 to OTnCKD0	Select OTMn count clock frequency division ratio. 00: No divide (Initial value) 01: Divide 2 of timer clock frequency 10: Divide 4 of timer clock frequency 11: Divide 8 of timer clock frequency
3	-	Reserved bit
2 to 0	OTnCK2～ OTnCK0	Select a timer clock source of OTMn. 000: LSCLK0(initial value) 001: HSCLK 010: HSOCLK 011: Reserved bit 100: External clock 0 input (EXTRG0) 101: External clock 1 input (EXTRG1) 110: External clock 2 input (EXTRG2) 111: External clock 3 input (EXTRG3)

[Note]

- **Set OTnCLK register when counter operation is stopped.**

9.2.13 OTMn Counter control trigger register 0 register (OTnCTRG0 : n=0 to 5)

OTnCTRG0 is a SFR to set OTMn counter control function.
The counter is controlled by taking Logical OR of the setting of this register with the setting of OTnCTRG1.
The register setting is enabled when OTnTGEN bit of OTCSTAT register is "1".

Address: 0xF414(OT0CTRG0L/OT0CTRG0), 0xF415(OT0CTRG0H),
0xF434(OT1CTRG0L/OT1CTRG0), 0xF435(OT1CTRG0H),
0xF454(OT2CTRG0L/OT2CTRG0), 0xF455(OT2CTRG0H),
0xF474(OT3CTRG0L/OT3CTRG0), 0xF475(OT3CTRG0H),
0xF494(OT4CTRG0L/OT4CTRG0), 0xF495(OT4CTRG0H),
0xF4B4(OT5CTRG0L/OT5CTRG0), 0xF4B5(OT5CTRG0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnCTRG0															
Byte	OTnCTRG0H								OTnCTRG0L							
Bit	OTnTRM02	OTnTRM01	OTnTRM00	-	OTnSTS03	OTnSTS02	OTnSTS01	OTnSTS00	OTnCAPEN0	-	-	-	OTnSPC0	OTnSP0	OTnSTC0	OTnST0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	OTnTRM02 to OTnTRM00	Select edge or level of OTMn counter control signal 0. If OTMnTRG and WDTCLK are selected, this bit is disabled. Both counter start and counter stop are fixed to rising edge. Other than above, the edge of both Counter start/Counter stop is fixed to rising edge. Counter start / Counter stop 000: Rising edge / Falling edge (initial value) 001: Falling edge / Rising edge 010: Rising edge / Falling edge 011: Falling edge / Falling edge 1X0: "H" level / "L" level 1X1: "L" level / "H" level X: either 1 or 0
12	-	Reserved bit
11 to 8	OTnSTS03 to OTnSTS00	Select the source of OTMn counter control signal 0. When selecting ORMnTRG or OTMn0/OTMn1, select a channel other than the same channel (e.g When OTM0 setting, select other than OTM0TRG, OTM00, or OTM01). When selecting OTMnTRG or OTMn0/OTMn1, select a channel other than the same channel (other than OTM0TRG, OTM00, or OTM01 for OTM0 settings). ・CAPTURE, PWM mode 0000: External trigger 0 input (EXTRG0) (initial value) 0001: External trigger 1 input (EXTRG1) 0010: External trigger 2 input (EXTRG2) 0011: Analog comparator CH0 0100: Analog comparator CH1 0101: Analog comparator CH2 0110: OTM00 0111: OTM01 1000: OTM10 1001: OTM11 1010: OTM20 1011: OTM0TRG (operational timer 0 trigger) 1100: OTM3TRG (operational timer 3 trigger) 1101: OTM4TRG (operational timer 4 trigger) 1110: OTM5TRG (operational timer 5 trigger) 1111: WDTCLK * For the information about OTMnTRG, see OTnINTEH in OTnINTE register

Bit No.	Bit symbol name	Description
7	OTnCAPEN0	Select whether to use the source of the counter control signal 0 set in OTnSTS03 to 00 bit as the trigger in capture mode. 0: Capture trigger disable (initial value) 1: Capture trigger enable
6 to 4	-	reserved bit
3	OTnSPC0	Select whether to clear the counter when the counter stop signal of counter control signal 0 occurs. The setting of this bit is enabled regardless of OTnSP0 bit setting. And if a register update request has been occurred using OTCUDn bit of OTCUD register, register is also updated when the counter stop signal of counter control signal 0 is occurred. 0: Counter clear disable (initial value) 1: Counter clear enable However, in the following cases, the counter is not cleared/registered when the counter stop signal of the counter control signal 0 is occurred. ・In case the counter is stopped by setting OTnTRM02 to 00 to "000" or "011". ・In case the counter is stopped by setting OTMnTRG with OTnSTS03 to 00. Regardless of the setting of OTnSTSYN bit of OTnMOD register, if the setting of this bit is "1", the counter is cleared when a counter stop signal of counter control signal 0 is occurred.
2	OTnSP0	Select enable/disable of the counter stop by counter control signal "0". 0: Counter stop disable (initial value) 1: Counter stop enable
1	OTnSTC0	Select whether to clear the counter when the counter start signal of counter control signal 0 occurs. The setting of this bit is enabled regardless of OTnST0 bit setting. And if a register update request has been occurred using OTCUDn bit of OTCUD register, register is also updated when the counter start signal of counter control signal 0 is occurred. 0: Counter clear disable (initial value) 1: Counter clear enable However, in the following cases, the counter is not cleared/registered when the counter start signal of the counter control signal 0 is occurred. ・In case the counter is in operation by setting OTnTRM02 to 00 to "000" or "011". ・In case the counter is in operation by setting ORMnTRG with OTnSTS03 to 00. Regardless of the setting of OTnSTSYN bit of OTnMOD register, if the setting of this bit is "1", the counter is cleared when the counter start signal of counter control signal 0 is occurred.
0	OTnST0	Select enable/disable of the counter start by counter control signal "0". 0: Counter start disable (initial value) 1: Counter start enable

[Note]

- **Trigger pulse width must be 2cloks wide of timer clock or more.**
- **In one-shot mode, set OTnTRM02 to OTnTRM00 bits to "1X0" or "1X1", and if the start condition and the level of the counter control signal 0 match, and the stop condition in one-shot mode is satisfied, the counting operation is continued (count-up is resumed from 0).**
- **If OTnCTRG0 register is set when a trigger event is enabled, a trigger event may occur at the moment of the setting. So when setting OTnCTRG0 register, verify that OTnTGEN bit of OTCSTAT register is "0" and change the setting of OTnCTRG0 register in the trigger operation stopped state.**

9.2.14 OTMn Counter control trigger 1 register (OTnCTRG1 : n=0 to 5)

OTnCTRG1 is a SFR to set OTMn counter control function.
The counter is controlled by taking Logical OR of the setting of this register with the setting of OTnCTRG0.
The register setting is enabled when OTnTGEN bit of OTCSTAT register is "1".

Address: 0xF3A0(OT0CTRG1L/OT0CTRG1), 0xF3A1(OT0CTRG1H),
0xF3A2(OT1CTRG1L/OT1CTRG1), 0xF3A3(OT1CTRG1H),
0xF3A4(OT2CTRG1L/OT2CTRG1), 0xF3A5(OT2CTRG1H),
0xF3A6(OT3CTRG1L/OT3CTRG1), 0xF3A7(OT3CTRG1H),
0xF3A8(OT4CTRG1L/OT4CTRG1), 0xF3A9(OT4CTRG1H),
0xF3AA(OT5CTRG1L/OT5CTRG1), 0xF3AB(OT5CTRG1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnCTRG1															
Byte	OTnCTRG1H								OTnCTRG1L							
Bit	OTnTRM12	OTnTRM11	OTnTRM10	-	OTnSTS13	OTnSTS12	OTnSTS11	OTnSTS10	OTnCAPEN1	-	-	-	OTnSPC1	OTnSP1	OTnSTC1	OTnST1
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	OTnTRM12 to OTnTRM10	Select edge or level of OTMn counter control signal 1. It is valid only when OTnSTS13 to OTnSTS10 bits of OTnCTRG1 register are selected for external triggering or analog comparator output, OTMn0, and OTMn1. Other than above, the edge of both Counter start/Counter stop is fixed to rising. Counter start / Counter stop 000: Rising edge / Rising edge (initial value) 001: Falling edge / Rising edge 010: Rising edge / Falling edge 011: Falling edge / Falling edge 1X0: "H" level / "L" level 1X1: "L" level / "H" level X: either 0 or 1
12	-	Reserved bit
11 to 8	OTnSTS13 to OTnSTS10	Select the source of OTMn counter control signal 1. When selecting ORMnTRG or OTMn0/OTMn1, select a channel other than the same channel (e.g When OTM4 setting, select other than OTM4TRG, OTM40, or OTM41). ・CAPTURE, PWM mode 0000: External trigger 0 input (EXTRG0)(initial value) 0001: External trigger 1 input (EXTRG1) 0010: External trigger 2 input (EXTRD2) 0011: Analog comparator CH0 0100: Analog comparator CH1 0101: Analog comparator CH2 0110: OTM30 0111: OTM40 1000: OTM41 1001: OTM50 1010: OTM51 1011: OTM0TRG (operational timer 0 trigger) 1100: OTM1TRG (operational timer 1 trigger) 1101: OTM2TRG (operational timer 2 trigger) 1110: OTM3TRG (operational timer 3 trigger) 1111: 16bit timer 0 trigger (TMH0TRG) *OTMnTRG can set the output timing with OTnIOOT0R/OTnIOTT0R/OTnIOH0R/OTnIO1F/OTnIO1R/OTnIO0F/OTnIO0F/OTnIOP bits of OTMn interrupt permit register (OTnINTE).

Bit No.	Bit symbol name	Description
7	OTnCAPEN1	Select whether to use the source of the counter control signal 1 set in OTnSTS13 to 10 bit as the trigger in capture mode. 0: Capture trigger disable (initial value) 1: Capture trigger enable
6 to 4	-	Reserved bit
3	OTnSPC1	Select whether to clear the counter when the counter stop signal of counter control signal 1 occurs. The setting of this bit is enabled regardless of OTnSP1 bit setting. And if a register update request has been occurred using OTCUDn bit of OTCUD register, register is also updated when the counter stop signal of counter control signal 1 is occurred. 0: Counter clear disable (initial value) 1: Counter clear enable However, in the following cases, the counter is not cleared/registered when the counter stop signal of the counter control signal 0 is occurred. ・In case the counter is stopped by setting OTnTRM12 to 10 to "000" or "011". ・In case the counter is stopped by setting OTMnTRG with OTnSTS13 to 10. Regardless of the setting of OTnSTSYN bit of OTnMOD register, if the setting of this bit is "1", the counter is cleared when the counter stop signal of counter control signal 1 is occurred.
2	OTnSP1	Select enable/disable of the counter stop by counter control signal "1". 0: Counter stop disable (initial value) 1: Counter stop enable
1	OTnSTC1	Select whether to clear the counter when the counter start signal of the counter control signal 1 occurs. The setting of this bit is enabled regardless of OTnST1 bit setting. And if a register update request has been occurred by OTCUDn bit of OTCUD register, register is also updated when the counter start signal of counter control signal 1 is occurred. 0: Counter clear disable (initial value) 1: Counter clear enable However, in the following cases, the counter is not cleared/registered when the counter start signal of the counter control signal 0 is occurred. ・In the case of the counter is in operation by setting OTnTRM12 to 10 to "000" or "011". ・In the case of the counter is in operation by setting ORMnTRG with OTnSTS13 to 10. Regardless of the setting of OTnSTSYN bit of OTnMOD register, if the setting of this bit is "1", the counter is cleared when the counter start signal of the counter control signal 1 is occurred.
0	OTnST1	Select enable/disable of the counter start by counter control signal "1". 0: Counter start disable (initial value) 1: Counter start enable

[Note]

- **Trigger pulse width must be 2cloks wide of timer clock or more.**
- **In one-shot mode, OTnTRM02 to OTnTRM00 bits are set to "1X0" or "1X1", and if the start condition and the level of the counter control signal 0 match, and the stop condition in one-shot mode is satisfied, the counting operation is continued (count-up is resumed from 0).**
- **If OTnCTRG0 register is set when a trigger event is enabled, a trigger event may occur at the moment of the setting. So, when setting OTnCTRG0 register, verify that OTnTGEN bit of OTCSTAT register is "0" and change the setting of OTnCTRG0 register in the trigger operation stopped state.**

9.2.15 OTMn Forced stop control trigger 0 register (OTnFTRG0 : n=0 to 5)

OTnFTRG0 is a SFR to set OTMn trigger function.
The counter is controlled by taking Logical OR of the setting of this register with the setting of OTnCTRG1.

Address: 0xF416(OT0FTRG0L/OT0FTRG0), 0xF417(OT0FTRG0H),
0xF436(OT1FTRG0L/OT1FTRG0), 0xF437(OT1FTRG0H),
0xF456(OT2FTRG0L/OT2FTRG0), 0xF457(OT2FTRG0H),
0xF476(OT3FTRG0L/OT3FTRG0), 0xF477(OT3FTRG0H),
0xF496(OT4FTRG0L/OT4FTRG0), 0xF497(OT4FTRG0H),
0xF4B6(OT5FTRG0L/OT5FTRG0), 0xF4B7(OT5FTRG0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnFTRG0															
Byte	OTnFTRG0H								OTnFTRG0L							
Bit	-	OTnFCE01	OTnFCE00	-	-	OTnEST02	OTnEST01	OTnEST00	-	-	-	-	-	OTnFCC0	OTnFCN01*	OTnFCN00
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14 to 13	OTnFCE01 to OTnFCE00	Select edge or level of OTMn Forced stop signal 0 00: Rising edge (initial value) 01: Falling edge 10: "H" level 11: "L" level
12 to 11	-	Reserved bit
10 to 8	OTnEST02 to OTnEST00	Select the source of OTMn Forced stop signal 0. 000: External trigger 0 input (EXTRG0) (initial value) 001: External trigger 2 input (EXTRG2) 010: Analog comparator CH0 011: Analog comparator CH1 100: Analog comparator CH2 101: OTM00 110: OTM10 111: OTM20
7 to 3	-	Reserved bit
2	OTnFCC0	Select whether to continue the counter operation when OTMn is forced stopped by forced stop signal 0 when forced stop is enabled. 0: When forced stopped, counter operation continues (initial value) 1: When forced stopped, counter operation is stopped. When OTnFCC0 bit is set to "1", OTnFCN00/OTnFCN01 bit must also be set to "1". When OTnFCN00/OTnFCN01 bits are set to "0" and shift to the forced stop state, unexpected short pulse signal might be output to OTOnA/OTOnB pin.
1	OTnFCN01	Select enable/disable of OTMn1 output forced stop. * Not available in CH2 and CH3. When the output force stop of OTMn1 is enabled, the output of OTMn1 is fixed to the output state selected by OTnSTPON bit of OTnMOD when the forced stop state is forced to stop due to the forced stop signal 0. If OTnFCC0 bit is set to "0" and select edge as forced stop trigger, forced stop state is released at the end of period. 0: Output forced stop disable (Initial value) 1: Output forced stop enable

Bit No.	Bit symbol name	Description
0	OTnFCN00	Select enable/disable of OTMn0 output forced stop. When the output forced stop of OTMn0 is enabled, the output of OTMn0 is fixed to the output state selected by OTnSTPOP bit of OTnMOD when the forced stop state is caused by a stop signal of 0. If OTnFCC0 bit is set to "0" and select edge as forced stop trigger, forced stop state is released at the end of period. 0: Output forced stop disable (Initial value) 1: Output forced stop enable

[Note]

- **When changing the setting of OTnFTRG0 register during counter operation, set the following bits of the register to "0" once, and then change the setting value.**
  - **OTnFCC0 bit**
  - **OTnFCN01 bit**
  - **OTnFCN00 bit**

9.2.16 OTMn Forced stop control trigger 1 register（OTnFTRG1：n=0 to 5）

OTnFTRG1 is a SFR to set OTMn trigger function.
The logical OR of the forced stop signal 0 set by OTnFTRG0 register and the forced stop signal 1 set by this register is input as the forced stop trigger.

Address: 0xF418(OT0FTRG1L/OT0FTRG1), 0xF419(OT0FTRG1H),
0xF438(OT1FTRG1L/OT1FTRG1), 0xF439(OT1FTRG1H),
0xF458(OT2FTRG1L/OT2FTRG1), 0xF459(OT2FTRG1H),
0xF478(OT3FTRG1L/OT3FTRG1), 0xF479(OT3FTRG1H),
0xF498(OT4FTRG1L/OT4FTRG1), 0xF499(OT4FTRG1H),
0xF4B8(OT5FTRG1L/OT5FTRG1), 0xF4B9(OT5FTRG1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnFTRG1															
Byte	OTnFTRG1H								OTnFTRG1L							
Bit	-	OTnFCE11	OTnFCE10	-	-	OTnEST12	OTnEST11	OTnEST10	-	-	-	-	-	OTnFCC1	OTnFCN11*	OTnFCN10
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14 to 13	OTnFCE11 to OTnFCE10	Select edge or level of forced stop trigger. 00: Rising edge (initial value) 01: Falling edge 10: "H" level 11: "L" level
12 to 11	-	Reserved bit
10 to 8	OTnEST12 to OTnEST10	Select a source of forced trigger signal 1. 000: External trigger 1 input (EXTRG1) (initial value) 001: External trigger 3 input (EXTRG3) 010: Analog comparator CH0 011: Analog comparator CH1 100: Analog comparator CH2 101: OTM30 110: OTM40 111: OTM50
7 to 3	-	Reserved bit
2	OTnFCC1	Select whether to continue the counter operation when OTMn is forced stopped by forced stop signal 1 when forced stop is enabled. 0: When forced stopped, counter operation continues (initial value) 1: When forced stopped, counter operation is stopped When OTnFCC1 bit is set to "1", OTnFCN10/OTnFCN11 bit must also be set to "1". When OTnFCN10/OTnFCN11 bits are set to "0" and shift to the forced stop state, unexpected short pulse signal might be output to OTOnA/OTOnB pin.
1	OTnFCN11	Select enable/disable of OTMn1 output forced stop. * Not available in CH2 and CH3. When the output force stop of OTMn1 is enabled, the output of OTMn1 is fixed to the output state selected by OTnSTPON bit of OTnMOD when the forced stop state is forced to stop due to the forced stop signal 1. If OTnFCC1 bit is set to "0" and select edge as forced stop trigger, forced stop state is released at the end of period. 0: Output forced stop disable (Initial value) 1: Output forced stop enable

Bit No.	Bit symbol name	Description
0	OTnFCN10	Select enable/disable of OTMn0 output forced stop. When the output forced stop of OTMn0 is enabled, the output of OTMn0 is fixed to the output state selected by OTnSTPOP bit of OTnMOD when the forced stop state is caused by a stop signal of 1. If OTnFCC0 bit is set to "0" and select edge as forced stop trigger, forced stop state is released at the end of period. 0: Output forced stop disable (Initial value) 1: Output forced stop enable

【注意】

- **When changing the setting of OTnFTRG1 register during counter operation, set the following bits of the register to "0" once, and then change the setting value.**
- **・OTnFCC1 bit**
- **・OTnFCN11 bit**
- **・OTnFCN10 bit**

9.2.17 OTMn interrupt enable register（OTnINTE：n=0 to 5）

OTnINTE is a SFR to control an interrupt and a trigger output of OTMn.
An interrupt is enabled when writing "1" to each bit of OTnINTEL register and notifies it to the interrupt controller.
A trigger output is enabled when writing "1" to each bit of OTnINTEH register and notifies it to the other channels.

Address: 0xF41A（OT0INTEL/OT0INTE），0xF41B（OT0INTEH），
0xF43A（OT1INTEL/OT1INTE），0xF43B（OT1INTEH），
0xF45A（OT2INTEL/OT2INTE），0xF45B（OT2INTEH），
0xF47A（OT3INTEL/OT3INTE），0xF47B（OT3INTEH）
0xF49A（OT4INTEL/OT4INTE），0xF49B（OT4INTEH），
0xF4BA（OT5INTEL/OT5INTE），0xF4BB（OT5INTEH）
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnINTE															
Byte	OTnINTEH								OTnINTEL							
Bit	-	-	OTnIOH0R	OTnIO1F *	OTnIO1R *	OTnIO0F	OTnIO0R	OTnIOP	-	OTnIETR	OTnIETS	OTnIE1F *	OTnIE1R *	OTnIE0F	OTnIE0R	OTnIEP
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Common explanation of each bit:
This bit enables or disables the appropriate interrupts and functions.
0: disable (initial value)
1: enable

Bit No.	Bit symbol name	Description (appropriate functions)
15	-	Reserved bit
14	-	Reserved bit
13	OTnIOH0R	CAPTURE mode: When set to enable, OTMnTRG is output when the value of OTnC register matches 1/2 of the value captured in OTn0R register. PWM mode: When set to enable, OTMnTRG is output when the value of OTnC register matches 1/2 of the value set in the buffer for OTn0R.
12	OTnIO1F	When set to enable, OTMnTRG is output when the value of OTnC register matches the value of the buffer for OTn1F. No output in CAPTURE mode.
11	OTnIO1R	When set to enable, OTMnTRG is output when the value of OTnC register matches the value of the buffer for OTn1R. No output in CAPTURE mode.
10	OTnIO0F	CAPTURE mode: When set to enable, OTMnTRG is output when captured to OTn0F register. PWM mode: When set to enable, OTMnTRG is output when the value of OTnC register matches the value of the buffer for OTn0F.
9	OTnIO0R	CAPTURE mode: When set to enable, OTMnTRG is output when captured to OTn0R register. PWM mode: When set to enable, OTMnTRG is output when the value of OTnC register matches the value of the buffer for OTn0R.
8	OTnIOP	When set to enable, OTMnTRG is output when the value of OTnC register matches the value of the buffer for OTnP.
7	-	Reserved bit

Bit No.	Bit symbol name	Description (appropriate functions)
6	OTnIETR	Trigger count start interrupt for OTMn An interrupt is output at the timing of the counter start triggered by OTMn. In addition, an interrupt is output at the timing of the counter clearing due to the occurrence of the trigger event of the counter start.
5	OTnIETS	Trigger count stop interrupt for OTMn An interrupt is output at the timing of the counter stop triggered by OTMn. In addition, an interrupt is output at the timing of the counter clearing due to the occurrence of the trigger event of the counter stop.
4	OTnIE1F	Interrupt by OTMn1 Fall Point When set to enable, OTMnINT is output when the value of OTnC register matches the value of the buffer for OTn1F. No output in CAPTURE mode.
3	OTnIE1R	Interrupt by OTMn1 Rise Point When set to enable, OTMnINT is output when the value of OTnC register matches the value of the buffer for OTnC. No output in CAPTURE mode.
2	OTnIE0F	Interrupt by OTMn0 Fall Point CAPTURE mode: When set to enable, OTMnINT is output when captured to OTn0F register. PWM mode: When set to enable, OTMnINT is output when the value of OTnC register matches the value of the buffer for OTn0F.
1	OTnIE0R	Interrupt by OTMn0 Rise Point CAPTURE mode: When set to enable, OTMnINT is output when captured to OTn0R register. PWM mode: When set to enable, OTMnINT is output when the value of OTnC register matches the value of the buffer for OTn0R.
0	OTnIEP	OTMn periodical interrupt When set to enable, OTMnINT is output when the value of OTnC register matches the value of the buffer for OTn0P.

9.2.18 OTMn interrupt status register（OTnINTS：n=0 to 5）

OTnINTS is a read only SFR to indicate OTMn interrupt status.
In addition to the reset function, bit 7 to 0 are initialized to "0" by writing "1" to the same bit of OTnINTC register

Address: 0xF41C（OT0INTSL/OT0INTS），0xF41D（OT0INTSH），
0xF43C（OT1INTSL/OT1INTS），0xF43D（OT1INTSH），
0xF45C（OT2INTSL/OT2INTS），0xF45D（OT2INTSH），
0xF47C（OT3INTSL/OT3INTS），0xF47D（OT3INTSH），
0xF49C（OT4INTSL/OT4INTS），0xF49D（OT4INTSH），
0xF4BC（OT5INTSL/OT5INTS），0xF4BD（OT5INTSH）
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTnINTS															
Byte	OTnINTSH								OTnINTSL							
Bit	-	-	-	-	-	-	-	-	OTnISFS	OTnISTR	OTnISTS	OTnIS1F *	OTnIS1R *	OTnIS0F	OTnIS0R	OTnISP
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Common explanation of each bit:
This bit indicates a status of an appropriate interrupt.
0: No appropriate interrupt occurs (initial value)
1: Appropriate interrupt occur

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 8	-	Reserved bit
7	OTnISFS	OTMn forced stop interrupt It is cleared by writing "1" to OTnICFS bit of OTnINTC register.
6	OTnISTR	OTMn Trigger counter start interrupt It is cleared by writing "1" to OTnICTR bit of OTnINTC register.
5	OTnISTS	OTMn Trigger counter stop interrupt It is cleared by writing "1" to OTnICTS bit of OTnINTC register.
4	OTnIS1F	Interrupt by OTMn1 Fall point Fixed to “0” at CAPTURE mode. It is cleared by writing "1" to OTnIC1F bit of OTnINTC register.
3	OTnIS1R	Interrupt by OTMn1 Rise point Fixed to “0” at CAPTURE mode. It is cleared by writing "1" to OTnIC1R bit of OTnINTC register.
2	OTnIS0F	Interrupt by OTMn0 Fall point In CAPTURE mode, the status of the capture data stored in OTn0F register is indicated. It is cleared by writing "1" to OTnIC0F bit of OTnINTC register. In CAPTURE mode, it is also cleared by reading OTn0F register.
1	OTnIS0R	Interrupt by OTMn0 Rise point In CAPTURE mode, the status of the capture data stored in OTn0R register is indicated. It is cleared by writing "1" to OTnIC0R bit of OTnINTC register. In CAPTURE mode, it is also cleared by reading OTn0R register.
0	OTnISP	OTMn periodical interrupt It is cleared by writing "1" to OTnICP bit of OTnINTC register.

9.2.19 OTMn interrupt clear register L/H (OTnINTCL, OTnINTCH : n=0 to 5)

OTnINTCL and OTnINTCH are SFRs to clear an interrupt status.
Writing "1" to bit 7 to 0 clears the interrupt status of the same bit in OTnINTS register.
When it is read, it always reads "0x0000".

Address: 0xF41E(OT0INTCL), 0xF41F(OT0INTCH), 0xF43E(OT1INTCL), 0xF43F(OT1INTCH),
0xF45E(OT2INTCL), 0xF45F(OT2INTCH), 0xF47E(OT3INTCL), 0xF47F(OT3INTCH),
0xF49E(OT4INTCL), 0xF49F(OT4INTCH), 0xF4BE(OT5INTCL), 0xF4BF(OT5INTCH)
Access: W
Access size: 8bit
initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	OTnINTCH								OTnINTCL							
Bit	OTnIR	-	-	-	-	-	-	-	OTnICFS	OTnICTR	OTnICTS	OTnIC1F *	OTnIC1R *	OTnIC0F	OTnIC0R	OTnICP
R/W	W	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* : Not available in CH2 and CH3.

Common explanation of each bit (bit5 to 0):
The bit clears an appropriate status.
"0" Writing : disable
"1" Writing : Clear an appropriate status

Bit No.	Bit symbol name	Description (appropriate functions)
15	OTnIR	OTMn interrupt request bit Write "1" before an interrupt routine completes "0"Writing : disable "1" Writing : If there is an outstanding interrupt source, output an interrupt request again.
14 to 8	-	Reserved bit
7	OTnICFS	OTMn forced stop interrupt status However, when the "H" level/"L" level is set by OTnFCE11 to10 bits of OTnFTRG0/OTnFTRG1 register, OTnISFS bit is not cleared when writing to OTnICFS bit It is automatically cleared when stop state caused by a forced stop trigger is released.
6	OTnICTR	OTMn Trigger counter start interrupt status
5	OTnICTS	OTMn Trigger counter stop interrupt status
4	OTnIC1F	OTMn 1 Fall Point interrupt status
3	OTnIC1R	OTMn 1 Rise Point interrupt status
2	OTnIC0F	OTMn 0 Fall Point interrupt status
1	OTnIC0R	OTMn 0 Rise Point interrupt status
0	OTnICP	OTMn periodical interrupt status

9.2.20 OTM common update register（OTCUD）

OTCUD is a SFR to update OTnP, OTn0R, OTn0F, OTn1R, and OTn1F register of OTMn during operation.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF380（OTCUD）
Access: W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								OTCUD							
Bit	-	-	-	-	-	-	-	-	-	-	OTCUD5	OTCUD4	OTCUD3	OTCUD2	OTCUD1	OTCUD0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 6	-	Reserved bits
5 to 0	OTCUD5 to OTCUD0	Write only bit to update OTnP, OTn0R, OTn0F, OTn1R, and OTn1F register of OTMn during operation. After updating the above register, by writing "1" to this bit, setting value of each register is transferred to the buffer at the end of the period, and pulse-width-modulated waveform is updated. OTnUD bit of OTnSTAT indicates update request status. "0" Writing : disable "1" Writing : update request occurs

9.2.21 OTM Common control register (OTCCON)

OTCCON is a SFR to set enable/disable OTMn output.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF382(OTCCONL/OTCCON), 0xF383(OTCCONH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTCCON															
Byte	OTCCONH								OTCCONL							
Bit	-	-	OT5SDN	OT4SDN	OT3SDN	OT2SDN	OT1SDN	OT0SDN	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bit
13 to 8	OT5SDN to OT0SDN	Set permission of OTMn0/OTMn1 output. ・CAPTURE mode Unused ・PWM mode 0: Output permission (default value) 1: Output inhibition (OTMn0/OTMn1 output fixed to "low" level)
7 to 0	-	Reserved bit

9.2.22 OTM Common start register (OTCSTR)

OTCSTR is a SFR to set valid/invalid of OTMn counter start.
This SFR commons to each channel. The digits in the bit symbol correspond to the channel number.
Writing to bits of non-equipped channels is not valid.

Address: 0xF384(OTCSTRL/OTCSTR), 0xF385(OTCSTRH)
Access: W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTCSTR															
Byte	OTCSTRH								OTCSTRL							
Bit	-	-	OT5ETG	OT4ETG	OT3ETG	OT2ETG	OT1ETG	OT0ETG	-	-	OT5STR	OT4STR	OT3STR	OT2STR	OT1STR	OT0STR
R/W	R	R	W	W	W	W	W	W	R	R	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bits
13 to 8	OT5ETG to OT0ETG	Enable count stop/start by trigger event. To disable, set by OTSCTP register. In the initial state after power-on, the trigger operation is disabled. "0" Writing : disable "1" Writing : Trigger operation (count stop/start) enabled
7 to 6	-	Reserved bits
5 to 0	OT5STR～ OT0STR	Start OTMn count by software. Count start by writing "1". In the initial state after power-on, count is stop. "0" Writing : disable "1" Writing : count start by software

9.2.23 OTM Common stop register（OTCSTP）

OTCSTP is a SFR to set enable/disable OTMn counter stop.
This SFR commons to each channel. The digits in the bit symbol correspond to the channel number.
Writing to bits of non-equipped channels is not valid.

Address: 0xF386（OTCSTPL/OTCSTP），0xF387（OTCSTPH）
Access: W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTCSTP															
Byte	OTCSTPH								OTCSTPL							
Bit	-	-	OT5DTG	OT4DTG	OT3DTG	OT2DTG	OT1DTG	OT0DTG	-	-	OT5STP	OT4STP	OT3STP	OT2STP	OT1STP	OT0STP
R/W	R	R	W	W	W	W	W	W	R	R	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bits
13 to 8	OT5DTG to OT0DTG	Disable count stop/start by a trigger event. To enable, set by OTSCTP register. In the initial state after power-on, the trigger operation is disabled. "0" Writing : disable "1" Writing : Trigger operation (count stop/start) disable
7 to 6	-	Reserved bits
5 to 0	OT5STP to OT0STP	Stop OTMn count by software. Count stop by writing "1" In the initial state after power-on, count is stop. "0" Writing : disable "1" Writing : count stop by software

9.2.24 OTM Common status register (TCSTAT)

TCSTAT is a SFR to indicate OTMn status.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF388(OTCSTATL/OTCSTAT), 0xF389(OTCSTATH)
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTCSTAT															
Byte	OTCSTATH								OTCSTATL							
Bit	-	-	OT5TGEN	OT4TGEN	OT3TGEN	OT2TGEN	OT1TGEN	OT0TGEN	-	-	OT5RUN	OT4RUN	OT3RUN	OT2RUN	OT1RUN	OT0RUN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bit
13 to 8	OT5TGEN to OT0TGEN	Verify setting status of OTMn. The initial state after power-on, the trigger operation is disabled. 0: Trigger operation (count stop/start) disable (initial value) 1: Trigger operation (count stop/start) enable
7 to 6	-	Reserved bit
5 to0	OT5RUN to OT0RUN	Indicate OTMn count status. OTnSTA bit of OTnSTAT register also indicate the same status. In the initial state after power-on, count is stop. 0: Counting stopped (default value) 1: Counting in operation

9.2.25 OTM Common break control register（OTCBRK）

OTCBRK is a SFR to set the behavior of OTMn during a break when a debugger is connected.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF38A（OTCBRK）
Access: R/W
Access size: 8bit
Initial value: 0x003F

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								OTCBRK							
Bit	-	-	-	-	-	-	-	-	-	-	OT5BRK	OT4BRK	OT3BRK	OT2BRK	OT1BRK	OT0BRK
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 6	-	Reserved bit
5 to 0	OT5BRK to OT0BRK	Set operation during the break of the operational timer during on-chip debugging, which is described in "28.4 Peripheral circuits to be operated during break during on-chip debugging" in "Chapter 28 On-chip debug function". When OTnBRK is set to "0", the counter operation of the channel set to "0" will continue even if the operational timer is unchecked during a break during on-chip debugging. If it is required to stop the operation by unchecking the function timer during a break during on-chip debugging, set OTnBRK bit corresponding to the channel you want to stop to "1". 0: Continues to operate during breaks. 1: When breaking, the debugger setting state is followed.

9.2.26 OTM common OTMn0 output control register (OTC0OCON)

OTC0OCON is a SFR that enables/disables the lower 8 bits (OTnOCONL register) of OTnOCON register.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF38C(OTC0OCONL/OTC0OCON), 0xF38D(OTC0OCONH)
Access: R/W
Access size: 8bit/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTC0OCON															
Byte	OTC0OCONH								OTC0OCONL							
Bit	-	-	OT50OEN1	OT40OEN1	OT30OEN1	OT20OEN1	OT10OEN1	OT00OEN1	-	-	OT50OEN0	OT40OEN0	OT30OEN0	OT20OEN0	OT10OEN0	OT00OEN0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bit
13	OT50OEN1	Set valid/invalid of OT5AND07 to 04 bit of OT5OCONL register 0: setting invalid (initial value) 1: setting valid
12	OT40OEN1	Set valid/invalid of OT4AND07 to 04 bit of OT4OCONL register 0: setting invalid (initial value) 1: setting valid
11	OT30OEN1	Set valid/invalid of OT3AND07 to 04 bit of OT3OCONL register 0: setting invalid (initial value) 1: setting valid
10	OT20OEN1	Set valid/invalid of OT2AND07 to 04 bit of OT2OCONL register 0: setting invalid (initial value) 1: setting valid
9	OT10OEN1	Set valid/invalid of OT1AND07 to 04 bit of OT1OCONL register 0: setting invalid (initial value) 1: setting valid
8	OT00OEN1	Set valid/invalid of OT0AND07 to 04 bit of OT0OCONL register 0: setting invalid (initial value) 1: setting valid
7 to 6	-	Reserved bit
5	OT50OEN0	Set valid/invalid of OT5AND03 to 00 bit of OT5OCONL register 0: setting invalid (initial value) 1: setting valid
4	OT40OEN0	Set valid/invalid of OT4AND03 to 00 bit of OT4OCONL register 0: setting invalid (initial value) 1: setting valid
3	OT30OEN0	Set valid/invalid of OT3AND03 to 00 bit of OT3OCONL register 0: setting invalid (initial value) 1: setting valid
2	OT20OEN0	Set valid/invalid of OT2AND03 to 00 bit of OT2OCONL register 0: setting invalid (initial value) 1: setting valid
1	OT10OEN0	Set valid/invalid of OT1AND03 to 00 bit of OT1OCONL register 0: setting invalid (initial value) 1: setting valid
0	OT00OEN0	Set valid/invalid of OT0AND03 to 00 bit of OT0OCONL register 0: setting invalid (initial value) 1: setting valid

9.2.27 OTM common OTMn1 output control register (OTC1OCON)

OTC1OCON is a SFR that enables/disables the upper 8 bits (OTnOCONH register) of OTnOCON register.
This SFR commons to each channel. Bit n corresponds to channel n.
Writing to bits of non-equipped channels is not valid.

Address: 0xF38E(OTC1OCONL/OTC1OCON), 0xF38F(OTC1OCONH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTC1OCON															
Byte	OTC1OCONH								OTC1OCONL							
Bit	-	-	OT51OEN1	OT41OEN1	-	-	OT11OEN1	OT01OEN1	-	-	OT51OEN0	OT41OEN0	-	-	OT11OEN0	OT01OEN0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 14	-	Reserved bit
13	OT51OEN1	Set valid/invalid of OT5AND17 to 14 bit of OT5OCONH register 0: setting invalid (initial value) 1: setting valid
12	OT41OEN1	Set valid/invalid of OT4AND17 to 14 bit of OT4OCONH register 0: setting invalid (initial value) 1: setting valid
11 to 10	-	Reserved bit
9	OT11OEN1	Set valid/invalid of OT1AND17 to 14 bit of OT1OCONH register 0: setting invalid (initial value) 1: setting valid
8	OT01OEN1	Set valid/invalid of OT0AND17 to 14 bit of OT0OCONH register 0: setting invalid (initial value) 1: setting valid
7 to 6	-	Reserved bit
5	OT51OEN0	Set valid/invalid of OT5AND13 to 10 bit of OT5OCONH register 0: setting invalid (initial value) 1: setting valid
4	OT41OEN0	Set valid/invalid of OT4AND13 to 10 bit of OT4OCONH register 0: setting invalid (initial value) 1: setting valid
3 to 2	-	Reserved bit
1	OT11OEN0	Set valid/invalid of OT1AND13 to 10 bit of OT1OCONH register 0: setting invalid (initial value) 1: setting valid
0	OT01OEN0	Set valid/invalid of OT0AND13 to 10 bit of OT0OCONH register 0: setting invalid (initial value) 1: setting valid

9.2.28 OTM Common cycle interlocking control register（OTCPC）

OTCPC is a SFR to select a value to set to OTnP.
When OTM2 and OTM3 are used in capture mode, the capture data is usable as OTMn cycle.
In this case, OTnP is update automatically with the following conditions.

- count value is stop at 0 (OTnC=0x0000)
- end of cycle
- counter clear by a trigger

When OTM2 and OTM3 are used PWM mode, they are updated in OTTM Common Update Register (OTCUD).
When OT20R and OT30R are selected, the value set in OTnP buffer is the value of OT20R and OT30R buffer.
In CAPTURE mode, OT20R and OT30R buffer are set to the value after capture.
In PWM mode, the value is set after OT20R are OT30R register is set and the update request is completed

This SFR commons to each channel.
Address: 0xF3F0（OTCPCL/OTCPC），0xF3F1（OTCPCH）
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	OTCPC															
Byte	OTCPCH								OTCPCL							
Bit	-	-	-	-	OT5PC1	OT5PC0	OT4PC1	OT4PC0	OT3PC1	OT3PC0	OT2PC1	OT2PC0	OT1PC1	OT1PC0	OT0PC1	OT0PC0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (appropriate functions)
15 to 12	-	Reserved bit
11 to 10	OT5PC0 to OT5PC1	Select a value to set to OT5P 00: writing value to OT5P (initial value) 01: OT20R 10: OT30R 11: setting prohibited
9 to 8	OT4PC0 to OT4PC1	Select a value to set to OT4P 00: writing value to OT4P (initial value) 01: OT20R 10: OT30R 11: setting prohibited
7 to 6	OT3PC0 to OT3PC1	Select a value to set to OT3P 00: writing value to OT3P (initial value) 01: OT20R 10: setting prohibited 11: setting prohibited
5 to 4	OT2PC0 to OT2PC1	Select a value to set to OT2P 00: writing value to OT2P (initial value) 01: setting prohibited 10: OT30R 11: setting prohibited
3 to 2	OT1PC0to OT1PC1	Select a value to set to OT1P 00: writing value to OT1P (initial value) 01: OT20R 10: OT30R 11: setting prohibited
1 to 0	OT0PC0 to OT0PC1	Select a value to set to OT0P 00: writing value to OT0P (initial value) 01: OT20R 10: OT30R 11: setting prohibited

[Note]

- **Set OTCPC register in a state where the counter operation is stopped.**
- **When the period is set to be linked with OT20R/OT30R, set the period of the count clock to be set to the linked OTMn so that it is less than or equal to the period of the count clock set in OTM2 and OTM3.**
- **If the period is set to longer than the period of the count clock set in OTM2 and OTM3, there is a possibility that an unintended value may be set to OTnP during period-linked operation.**

9.3 Description of operation

9.3.1 Common sequence (Initial settings common to all modes)

OTMn start operation by OTCSTR register after the setting 1 to 6 as follows.
During operation, it is available to check the hardware status such as interrupt status and update the period/event settings.

1: Mode setting (OTnMOD register)
Select CAPTURE/PWM mode by OTnMOD register.
In addition, set the continuous mode/one-shot mode, waveform output status when stopped, etc.

2: Clock setting (OTnCLK register)
Select timer clock and count clock. Set the source clock and its divider ratio.

3: Trigger setting (OTnCTRG0/1 register, OTnFTRG0/1 register)
OTnCTRG0/1 is set in the case of start or stop the counter by a trigger.
OTnFTRG0/1 is set in the case of there is a trigger that wants to be stopped in addition to the trigger of OTnCTRG0/1, such as stopping in the event of an abnormality.

Select a source and edge of an event trigger to start/stop counter and an action when a trigger is entered in OTnCTRG0 and OTnCTRG1 register.
Select a source of forced stop trigger and behavior at forced stop in OTnFTRG0, OTnFTRG1 registe.

4: Interrupt setting (OTnINTE register)
Set an interrupt source.
Select from cycle, event (Counter match, Capture), trigger start, and stop interrupt.

5: Period/Duty setting (OTnP register, OTn0R register, OTn0F register, OTn1R register, OTn1F register)
Set period, data for counter match, and etc.
The period can also be set in OTCPC/OTCADC register.

	CAPTURE mode	PWM mode
OTnP register	Period in continuous mode or timeout period in one-shot mode	
OTn0R register	(capture data)	OTMn0 output Duty
OTn0F register		
OTn1R register	（Unused）	OTMn1 output Duty
OTn1F register		

The formula for calculating the period is as follows:

$$T_{period} = \frac{OTnP + 1}{\text{Count clock frequency [Hz]}} \quad (OTnP : 0x0001 \sim 0xFFFF)$$

6: External output signal selecting

By OTnOCON register, set a combination of a logical AND of OTMn0/OTMn1 output with OTMn0/OTMn1 output of other channels, analog comparator output, and EXTRG input.
After the setting, enables OTnOCON register setting by OTC1OCON/OTC1OCON register.
OTnOSL1 and OTnOSL0 bits of OTnMOD register select whether to output OTMn0/OTMn1 output (OTMn0S/OTMn1S) from OTOnA/OTOnB pin.
In addition, select whether or not to invert OTMn0/OTMn1 output from OTOnA pin by OTnOSNA bit, select whether or not to invert OTMn0/OTMn1 output from OTOnB pin by OTnOSNB bit.
Fig. 9-2 shows the configuration of external output signal selection.

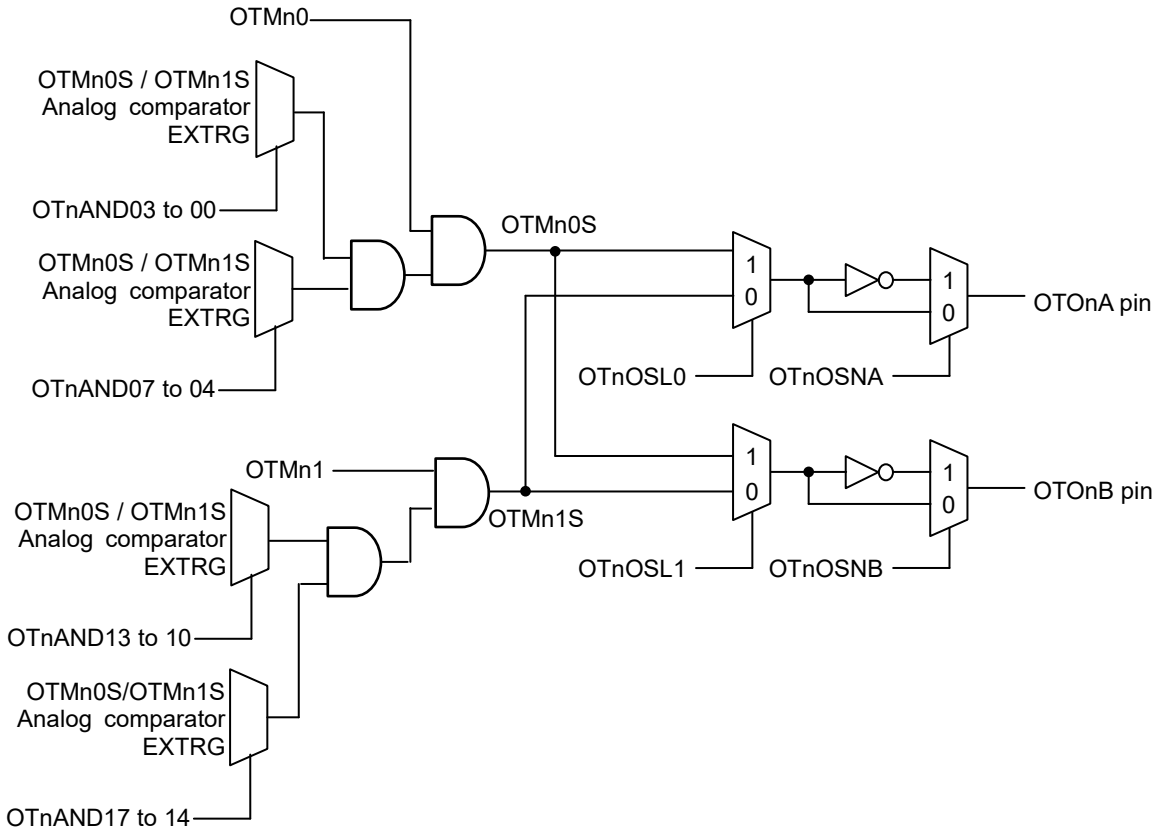


Fig. 9-2 Configuration of External Output Signal Selection

7: Start/Stop control

Count operation is started by software start or acceptance of a counter start/stop event trigger.
The counter counts up on the rising edge of the count clock.
Because software start/stop is synchronized by the timer clock, after a maximum of 4 cycles of the timer clock from the start, OTnSTA bit becomes "1" and the counter starts operating.
Since the software start/stop is synchronized with the timer clock, OTnSTA bit becomes "1" at a maximum of 4 cycles after the timer clock is set in the software, and the counter starts operating.
Stop case, the count stops by software stop and OTnSTA bit becomes "0". In this case, the count value is held.
If the software is started again, it will restart after a maximum of 4 cycles of the timer clock.
To clear the counter, write an arbitrary value to OTnC register.

8: Processing in operation

The status of counters and pins are monitorable during operation with OTnSTAT register, OTCSTAT register, and OTnINTS register.
In case of changing the waveform such as PWM, set a corresponding bit of OTCUD register after setting the period/event, and it will be updated at the next PWM cycle.
In addition, setting OTnSDN bit of OTCCON register forces OTMn0/OTMn1 output to be fixed at the "L" level.

9.3.2 Counter operation (Common to all modes)

The counters inside OTMn behaves the same operation in each mode.
Count up to OTnP register setting value.
In continuous mode (OTnOST bit of OTnMOD register is "0"), the counter is cleared at the time of overflow and continues the counting operation.
In one-shot mode (OTnOST bit of OTnMOD register is "1"), the counter is cleared at the time of overflow and stops the counting operation.
Counting operation start/stop is controllable by software or trigger events.

9.3.2.1 Count start/stop by software

When "1" is written to OTnSTR bit of OTCSTR register, OTnSTA bit of OTnSTAT register, which indicates the counting status, becomes "1" and the counting operation starts.
In one-shot mode (OTnOST bit of OTnMOD register is "1"), the counting operation is stopped due to overflow, and OTnSTA bit of OTnSTAT register, which indicates the counting status, is automatically set to "0".

When "1" is written to OTnSTP bit of OTCSTP register while the counter is running (OTnSTA bit of OTnSTAT register indicating the counting status is "1"), the counter will stop running.
When OTnSTSYN bit of OTnMOD register is "1", the count will continue from the end of the period to the value set by OTnDLY and then stop.
Confirm that the counter is stopped by setting OTnSTA bit of OTnSTAT register to "0" in the software.
When the counter is stopped, the value of the counter is held.
After the counter is stopped, when written "1" to OTnSTR bit of OTCSTR register again, the counter will continue to operate from the value at the time of stopping.
If the counter is to be cleared, a write operation is performed to OTnC register while it is stopped.
When restart the counter after that, write "1" to OTnSTR bit of OTCSTR register after confirming that OTnC register is "0x0000".

The timing at which the period/duty setting register setting is reflected to the buffer is as follows.
If cycle/duty register is written with counter is 0 with counter is stopped, it will be reflected at the start.
If cycle/duy register is written in timer operation, write is executed during timer operation, it will be reflected in the next PWM cycle of the update request with OTCUDn bit of OTCUD register.
Even if the timer is stopping, if the counter is not "0", even if a write is made, it will not be reflected until an update is requested. Reflect the setting value in one of the following ways.

- Reflect the setting value in one of the following ways. Execute a write operation to OTnC register and clear the counter.
  The settings are updated when start.
- After rewriting the corresponding register, update request with OTCUDn bit of OTCUD register.
  After starting, the settings are updated in the next cycle.

See "9.3.7 Changes during periodic and duty operation" for details on when the period/duty setting register is reflected.

9.3.2.2 Start/Stop by Trigger event

If write "1" to OTnETG bit of OTCSTR register, counter operation becomes controllable by the trigger event.
Trigger selection etc. is set by OTnCTRG0/OTnCTRG1 register.
Trigger event source is selectable from EXTRG0 to EXTRG3, TMH0INT, WDTCLK, OTMn0／OTMn1, OTMnTRG, and CMPnINT.
Depending on the trigger event that select, actions (counter start, counter stop, counter clear) is selectable.
When the counter is stopped due to a trigger event, if OTnSTSYN bit of OTnMOD register is "1", the count will continue from the end of the period to the value set by OTnDLY and then stop.

9.3.3 CAPTURE mode operation

In CAPTURE mode, the count value when the event trigger factor occurs is stored in OTn0R register or OTn0F register.
The event trigger factors to be captured are the same as those used for counter start/stop.

OTn0R register stored data	Count value when the rising edge of the event trigger occurs
OTn0F register stored data	Count value when the falling edge of the event trigger occurs

9.3.3.1 Example of CAPTURE mode operation

The following is an example of measuring only one period and duty of the PWM signal input from the EXTRG0 pin using the start/stop of a counter due to a trigger event.
Follow the steps below to set each register before measuring.

Step1: Select CAPTURE mode by writing "1" to OTnMD0 bit of OTnMOD register.
Step2: When an interrupt is used, set trigger counter stop interrupt to enable by writing "1" to OTnIE0R bit of OTnINTE register.
Step3: Setting OTnCTRG0 Register
Write OTnSTS03-00 = 0000 : Select EXTRG0 as trigger event
Write OTnST0 =1: Enabled to start by triggering the counter
Write OTnSP0=1: Counter trigger stop enabled
Write OTnCAPEN0=1: Capture trigger enabled
Write OTnTRM02 to 00 = 000: Counter start/stop is the rising edge of the trigger
Step4: Write "1" to OTnETG bit of OTCSTR register to enable the trigger operation of the capture.

An example of the operation in the case of the measurement in Figure 9-3 is as follows.

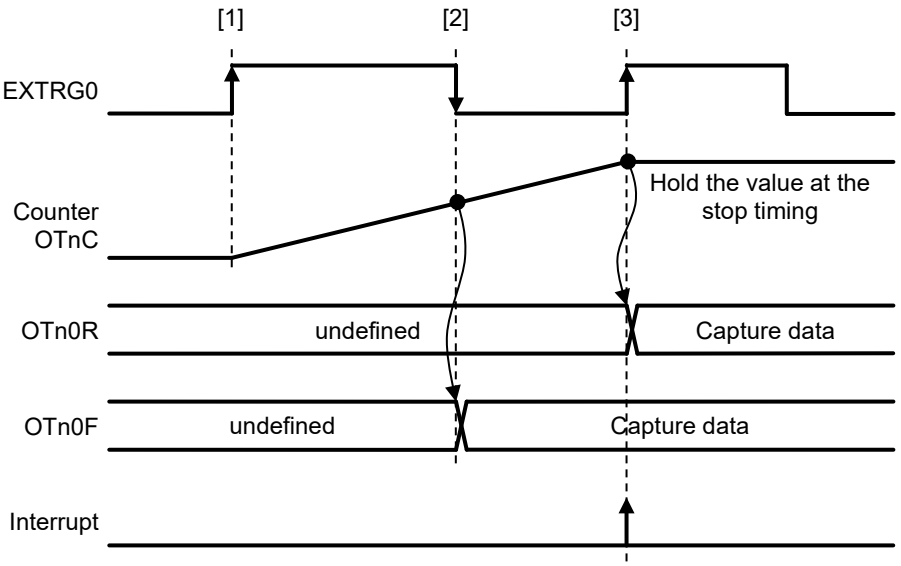


Fig. 9-3 Example of periodic/duty measurement of a PWM signal (only one period)

[1] Start counting when EXTRG0 rises
[2] When EXTRG0 fall timing, the value of OTMn counter register OTnC is stored in OTn0F register.
[3] When EXTRG0 rise timing, the value of OTMn counter register OTnC is stored in OTn0R register.
Stop counting, and an interrupt occurs.
The period of the PWM signal input from EXTRG0 pin is the value of OTn0R register, and the duty is the value of OTn0F register.
The following example shows how to measure the period and duty of a PWM signal (continuous) input from the EXTRG0 pin using the start/stop of a counter due to a trigger event.

Follow the steps below to set each register before measuring.

Step1: Write "1" to OTnMD0 bit of OTnMOD register to select CAPTURE mode.
Step2: When an interrupt is used, write "1" to OTnIE0R bit to enable 0 Rise Point interrupt.
Step3: Write "0000" to OTnSTS03 to OTnSTS00 bit of OTnCRTG0 register to set EXTRG0 as trigger event source.
Write "1" to OTnST0 bit to enable counter start function.
Write "1" to OTnSPC0 bit to enable counter clearing when a counter stop trigger event occurs.
Write "1" to OTnCAPEN0 bit to use trigger event source selected by OTnCTRG0 register as capture trigger.
Step4: Write "000" to OTnTRM02 to OTnTRM00 of OTnCTRG0 register to select a trigger by rising edge for both counter start and counter stop.
Step5: Write "1" to OTNnETG bit of OTCTRS register to enable capture trigger operation.

An example of the operation in the case of the measurement in Figure 9-4 is as follows.

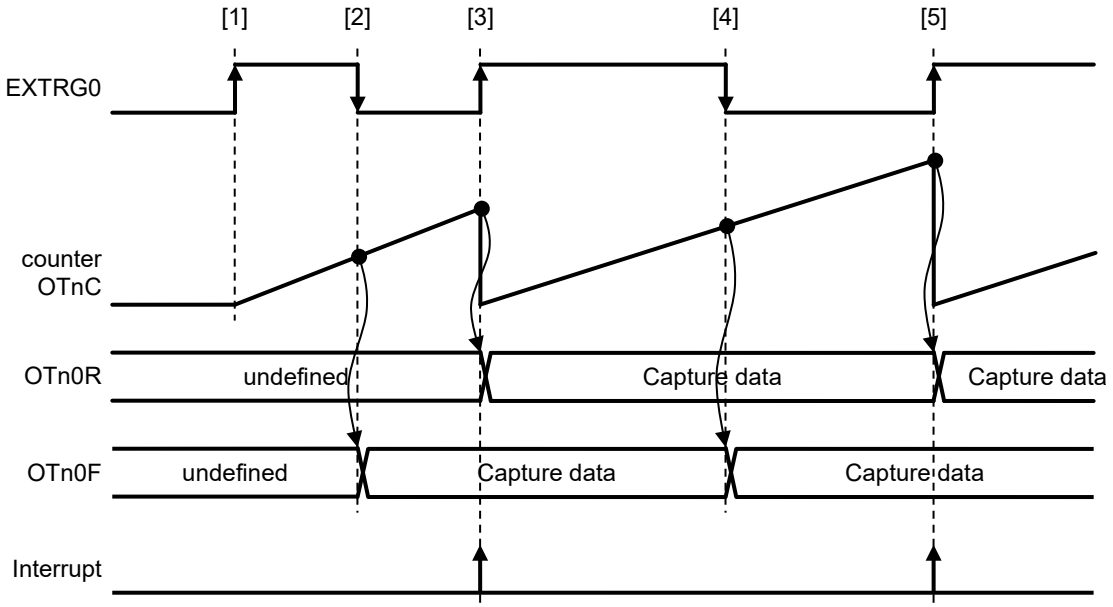
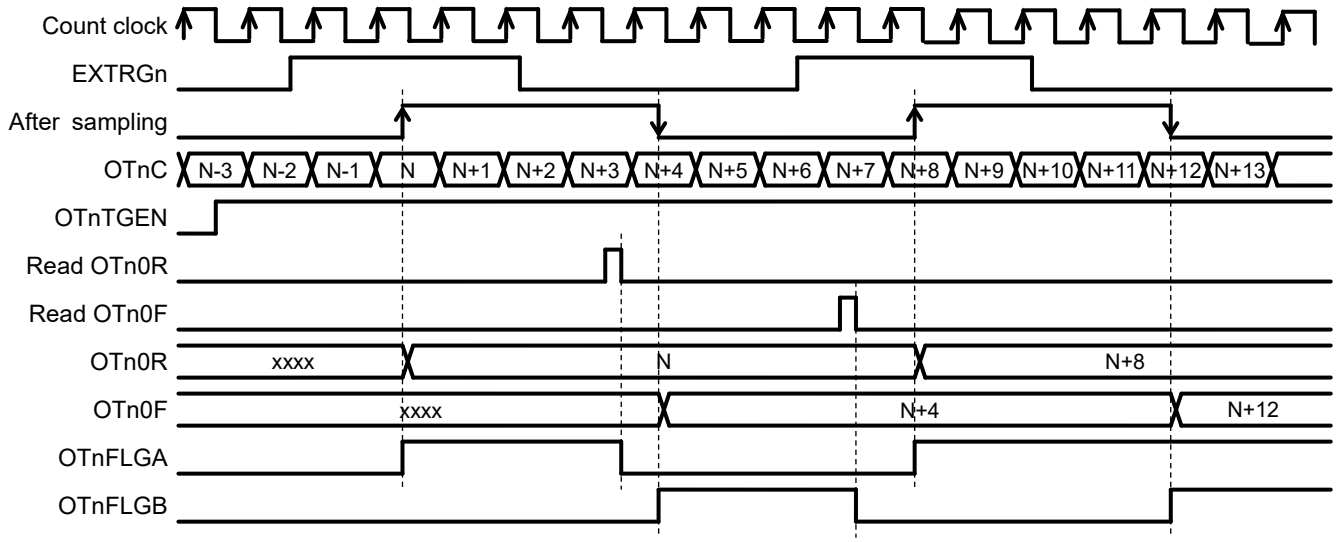


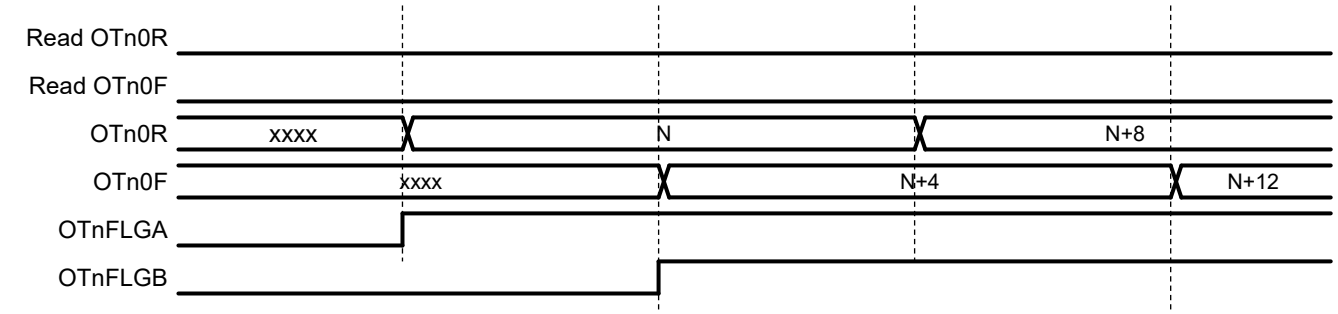
Fig. 9-4 Example of periodic/duty measurement of a PWM signal (continuous)

[1] Counter operation starts when EXTRG0 rises
[2] When EXTRG0 fall timing, the value of OTMn counter register OTnC is stored in OTn0F register.
[3] When EXTRG0 rise timing, the value of OTMn counter register OTnC is stored in OTn0R register.
Counter clear, 0 Rise Point interrupt occurred. After confirming the interrupt, clear the interrupt status.
Counter operation continues
[4] When EXTRG0 fall timing, the value of OTMn counter register OTnC is stored in OTn0F register.
[5] When EXTRG0 rise timing, the value of OTMn counter register OTnC is stored in OTn0R register.
Counter clear, 0 Rise Point interrupt occurred. After confirming the interrupt, clear the interrupt status.
Counter operation continues
The period of the PWM signal input from EXTRG0 pin is the value of OTn0R register, and the duty is the value of OTn0F register.

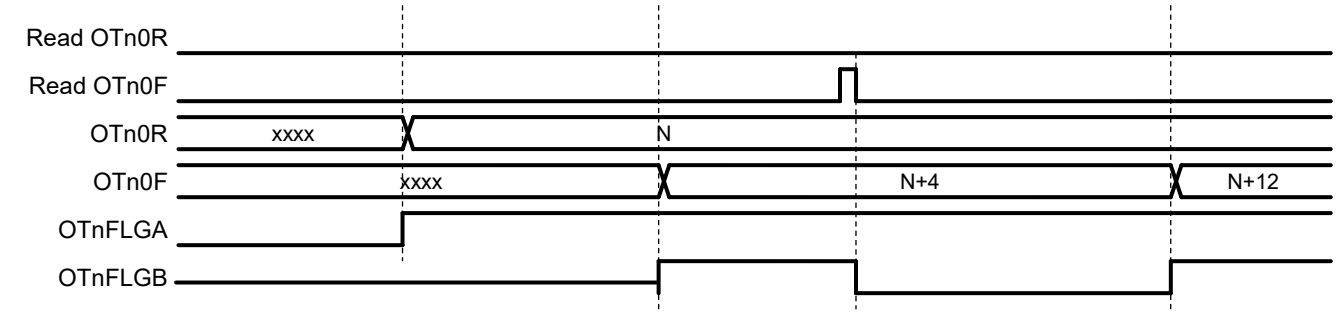
In addition, the behavior after capture depends on the setting of OTnOST bit in OTnMOD register.
In Auto mode (OTnOST="0")
When the signal comes up again and the counter is restarted, the value of OTn0R register is updated.
In Single mode (OTnOST="1")
When the signal comes up again and the counter is restarted, the value of OTn0R register is not updated.



(a) In case read out register value before next trigger is coming (common OTnOST=0,1)



(b) In case next trigger is coming without register read out (OTnOST=0)



(c) In case next trigger is coming without register read out (OTnOST=1)

Fig.9-5 CAPTURE mode operation timing

9.3.4 PWM mode operation

In PWM mode, generate a pulse with a period set by OTnP register.
Set the rise of OTMn0 output with OTn0R register and set the fall of OTMn0 output with OTn0F register.
Set the rise of OTMn1 output with OTn1R register and set the fall of OTMn1 output with OTn1F register.

9.3.4.1 PWM mode output waveform

In continuous mode, initial value of OTMn0/OTMn1 output is the level set by OTnSTPON/OTnSTPON bit of OTnMOD register.
If the count value matches the value of OTn0R/OTn1R, OTMn0/OTMn1 outputs "H".
If the count value matches the value of OTn0F/OTn1F, OTMn0/OTMn1 outputs "L".
Repeat above operation until OTMn is stop.
In one-shot mode, it will automatically stop and reach the level set by OTnSTPOP/OTnSTPON bits after one cycle.

Figure 9-6(a) shows the operating waveforms of OTMn0/OTMn1 in PWM mode (continuous mode), and Figure 9-6(b) shows the operating waveforms of OTMn0/OTMn1 in PWM mode (one-shot mode).

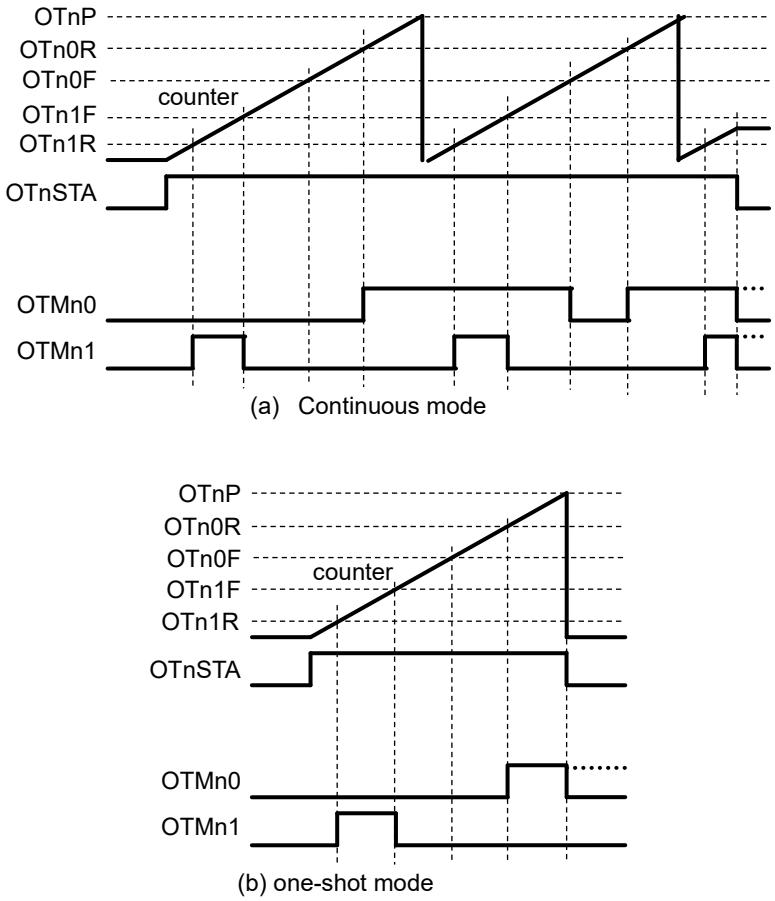


Fig. 9-6 OTMn0/OTMn1 waveform in PWM mode

By OTnOSL1/OTnOSL0 bit of OTnMOD register, the followings are selectable.
The phase of output fom OTOnA/OTOnB pin.
Whether or not OTOnA pin output is inverted by OTnOSNA bit.
Whether or not OTOnB pin output is inverted by OTnOSNB bit.

Figure 9-7-1 shows the output waveform in PWM mode when the phase output from OTOnA/OTOnB pin is set with OTnOSL1 and OTnOSL0 bits of OTnMOD register.
Figure 9-7-2 shows the output waveform in PWM mode when OTOnA pin output is inverted with OTnOSNA bit and OTOnB pin output is inverted with OTnOSNB bit.

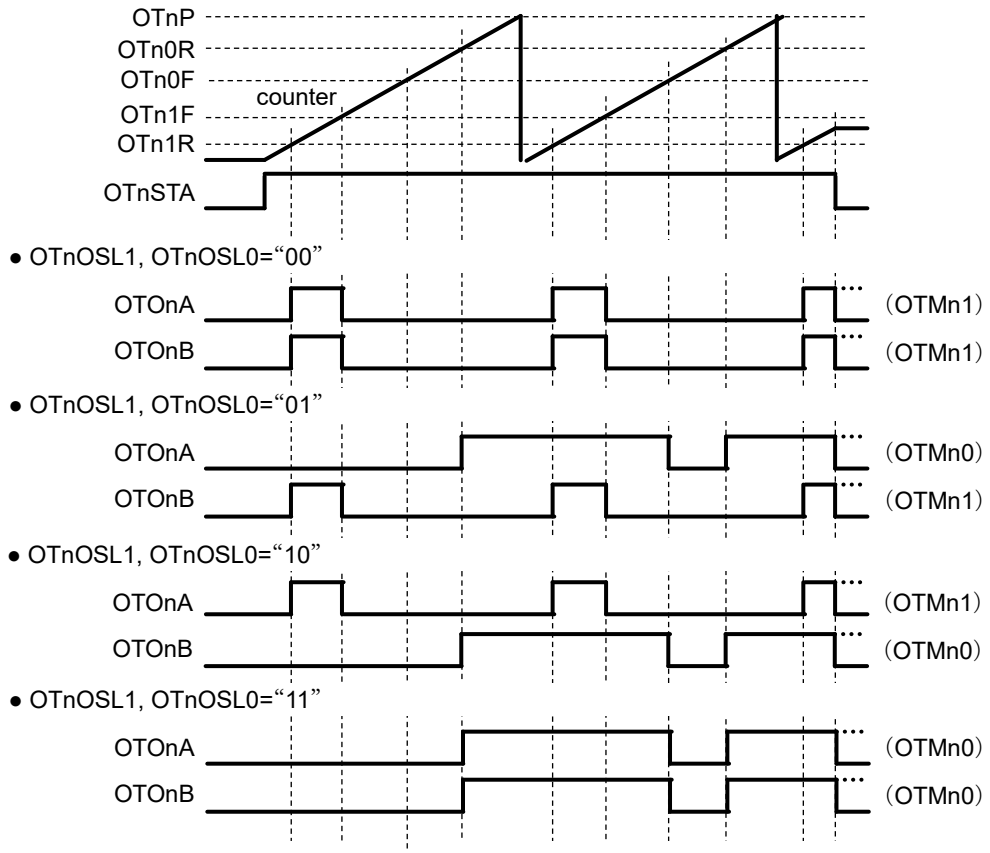


Fig. 9-7-1 PWM mode output waveform (Output phase selection by OTnOSL1 and OTnOSL0 bit)

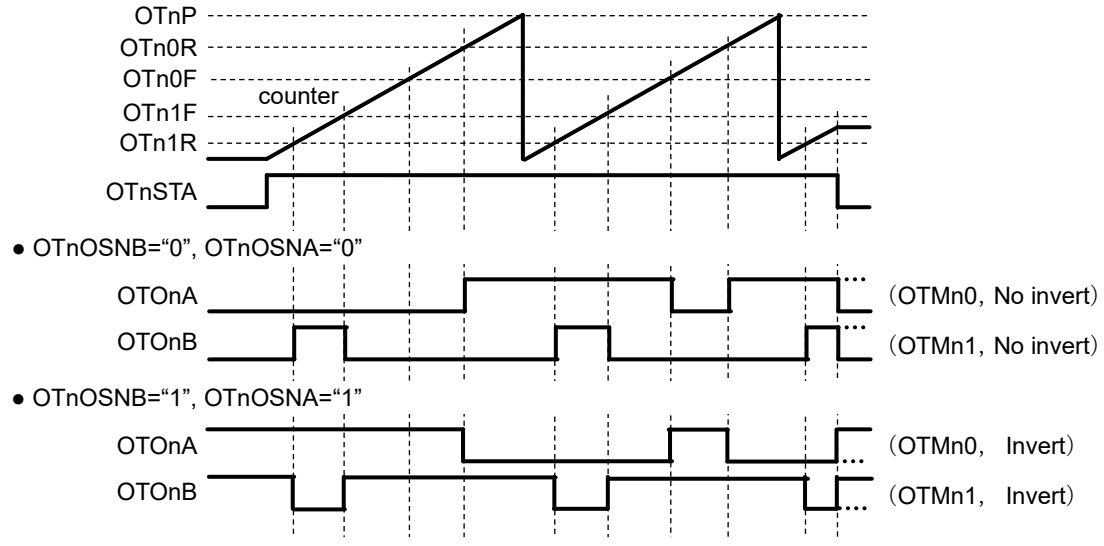


Fig. 9-7-2 PWM mode output waveform (Select with or without output inversion by OTnOSNB and OTnOSNA bits)

When the counter is stopped by software control or event triggering, or when the output waveform is fixed by forced stop the state of OTMn0/OTMn1 output can be set by OTnSTPOP/N bit of OTnMOD register.

1) Case of OTnSTPOP/N bit is "0"
OTMn0/OTMn1 output becomes "L" level when it is stopped.
If the count restart from the stopped state, OTMn0/OTMn1 output will remain "L" level output during the period and will change according to the count value from the next period.
When the output waveform is fixed by forced stop, "L" level output is held until the next period after the forced stop is released.

2) Case of OTnSTPOP/N bit is "1"
OTMn0/OTMn1 output becomes "H" level when it is stopped.
If the count restart from the stopped state, OTMn0/OTMn1 output will remain "H" level output during the period and will change according to the count value from the next period.
When the output waveform is fixed by forced stop, "H" level output is held until the next period after the forced stop is released.

When the counter is cleared by the trigger input, or when the counter is cleared by software after the counting operation is stopped, the count value is counted up from "0x0000" and the output changes according to the count value.

Fig. 9-8 shows the output waveform of PWM mode when the counter is stopped when OTnSTPOP/N bit of OTnMOD register is set to "0" and OTnSTPOP/N bit of OTnMOD register is set to "1".

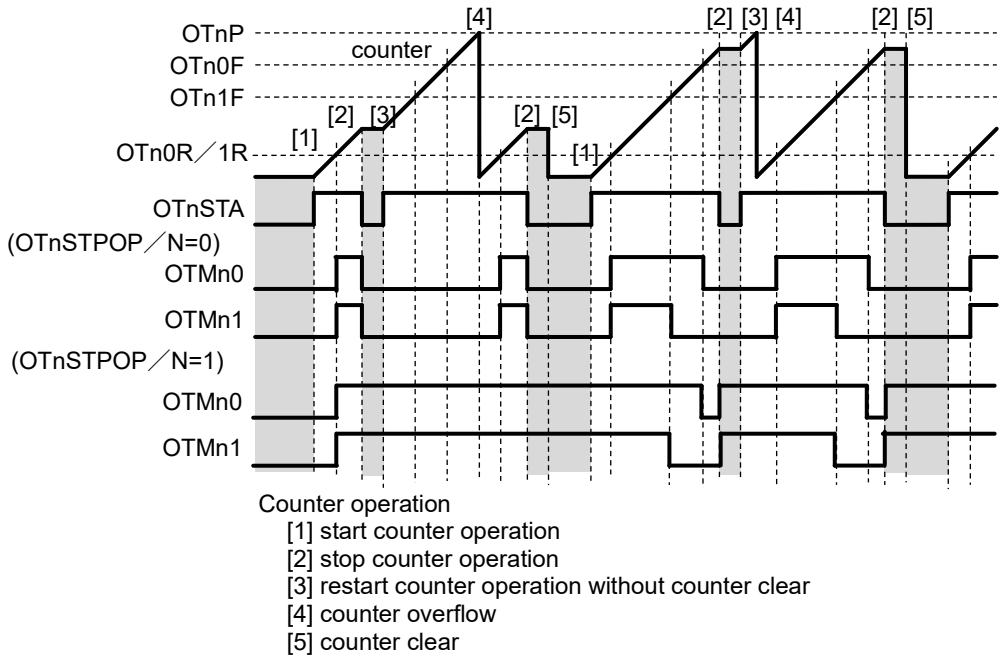


Fig. 9-8 PWM mode output waveform (counter is sopped)

By OTnOCON register, it is feasible to output a signal obtained by logical AND between OTMn1/OTMn0 output and OTMn1/OTMn0 output of another channel, or output of a comparator, external trigger input to OTOnA/OTOnB pin. The channel selection for logical AND is selectable in OTnAND17 to OTnAND10, OTnAND07 to OTnAND00.

Fig. 9-9-1 shows the output waveform of PWM mode setting by OT0OCON register when the logical AND of OTM00 and OTM10 is output to OTO0A pin and the logical AND of OTM01 and OTM11 is output to OTO0B pin .
Fig. 9-9-2 shows the output waveform of PWM mode setting by OT0OCON register when the logical AND of OTM00, OTM10, and OTM40 is output to OTO0A pin, and the logical AND of OTM01, OTM11, and OTM41 is output to OTO0B pin.

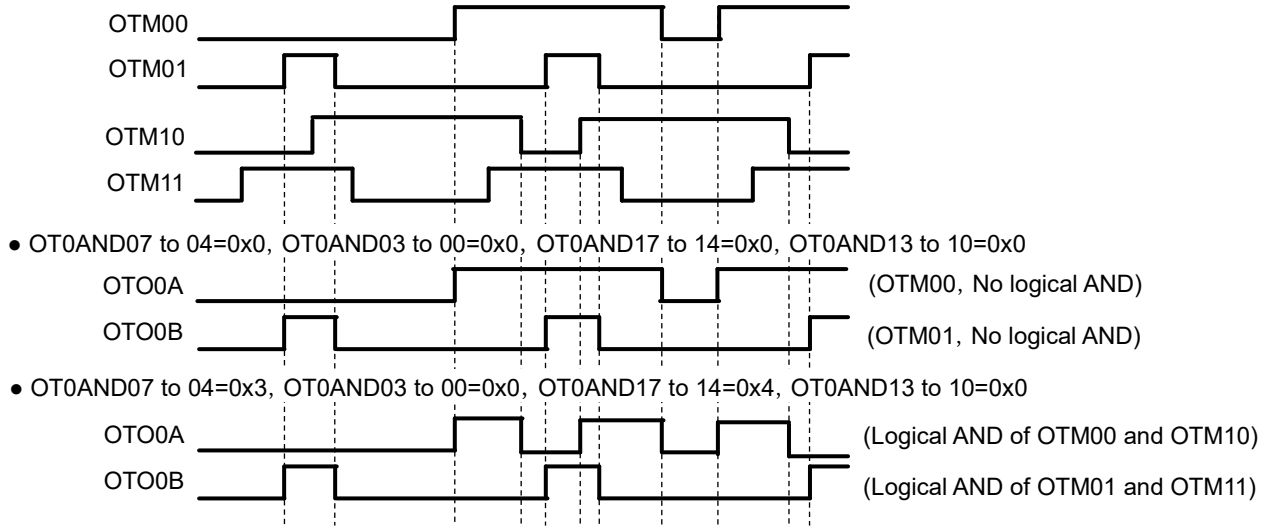


Fig. 9-9-1 PWM mode output waveform using OTnM output logic control register

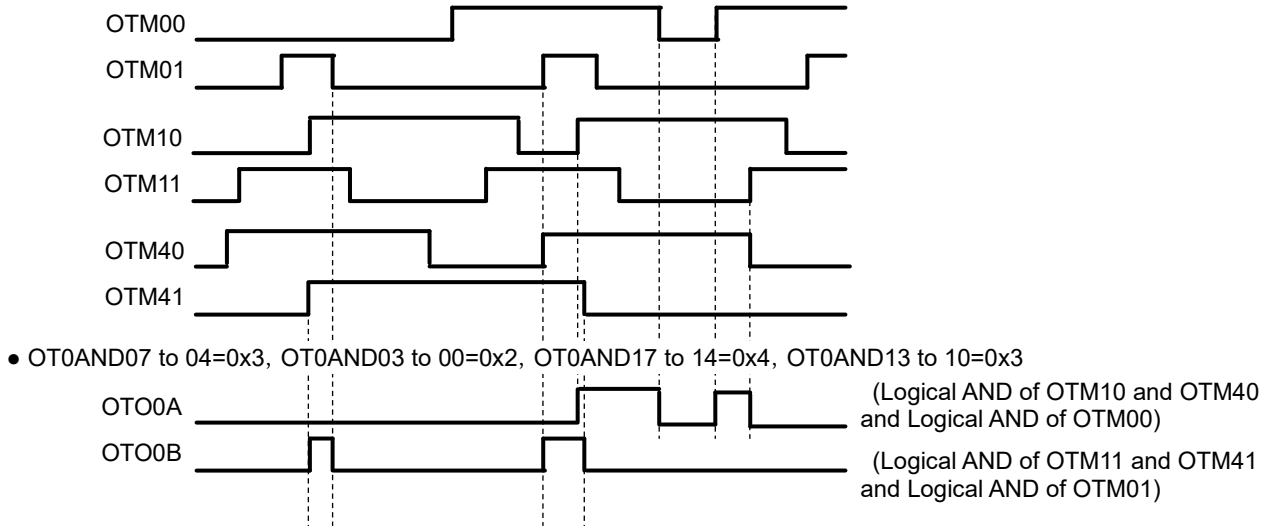


Fig. 9-9-2 PWM mode output waveform using OTnM output logic control register

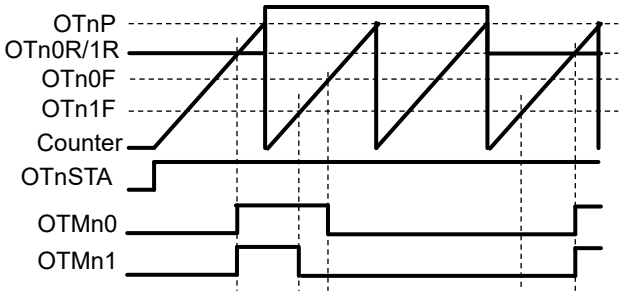
9.3.5 PWM duty setting

9.3.5.1 Output waveform control when Duty = 100%, 0%

The duty of the pulse width modulated output waveform are configurable to 100% or 0% by setting OTn0R, OTn0F, OTn1R, and OTn1F register in PWM mode.
If OTn0R(OTn1R) register is set to a value larger than OTnP's, or if OTn0R(OTn1R) register and OTn0F(OTn1F) register are set to the same value, the duty = 0%.
If OTn0F(OTn1F) register is set to a value larger than OTnP, the duty = 100%.

Fig. 9-10-1 shows the output waveform with duty = 0% in PWM mode, and Fig. 9-10-2 shows the duty = 100%.

	OTn0R/1R	OTn0F/1F
OTMn0	> OTnP	≤ OTnP
	*	= OTn0R
OTMn1	> OTnP	≤ OTnP
	*	= OTn1R



*: Don't care
Fig. 9-10-1 OTn0R/1R, OTn0F/1F register and OTMn0/OTMn1 output (Duty = 0%) in PWM mode

	OTn0R/1R	OTn0F/1F
OTMn0	≤ OTnP	> OTnP
OTMn1	≤ OTnP	> OTnP

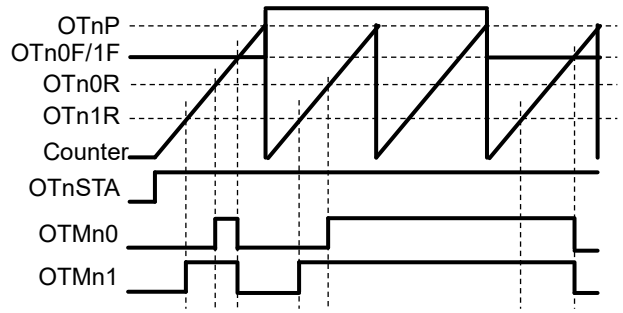


Fig. 9-10-2 OTn0R/1R, OTn0F/1F register and OTMn0/OTMn1 output (Duty = 100%)

9.3.6 Control of External clock input / Event trigger / Forced stop trigger

Operational timer is controllable by event input such as an external pin input as below.

- External clock input
  External pin input is usable as a counter clock.
  Set the clock by OTnCLK register.
- Event trigger
  Controllable counter start and stop control by events of an external pin input, analog comparator output, operational timer, and other channel output triggers.
- Forced stop trigger
  Forced stop control of counter and outputs by a trigger of an external pins input.

The external clock input is selectable from EXTRG0 to EXTRG3 to be used as a count clock.
Event triggers are used to start/stop of counter and to trigger captures.
A trigger source is selectable from EXTRG0 to EXTRG3, CMP0INT to CMP2INT, TMH0TRG, WDTCLK, OTMn0/OTMn1, and OTMnTRG.
The forced stop trigger is a trigger to stop OTMn0/OTMn1 output.
By setting OTnFCC0/OTnFCC1 bit of OTnFTRG0/OTnFTRG1 register to "1", the counter operation can be stopped during forced stop.
By setting OTnFCN01, OTnFCN00, OTnFCN11, and OTnFCN10 bits to "1", the output of OTMn0/OTMn1 will be the value set by OTnSTPON/OTnSTPOP bit of OTnMOD register during forced stop.
When OTnFCN01, OTnFCN00, OTnFCN11, and OTnFCN10 bits are set to "0", OTMn0/OTMn1 outputs hold their current state during forced stop.
A trigger source is selectable from EXTRG0 to EXTRG3, CMP0INT to CMP2INT, and OTMn0/OTMn1
The start/stop of the counter by event trigger is delayed by 2 clocks of the timer clock.

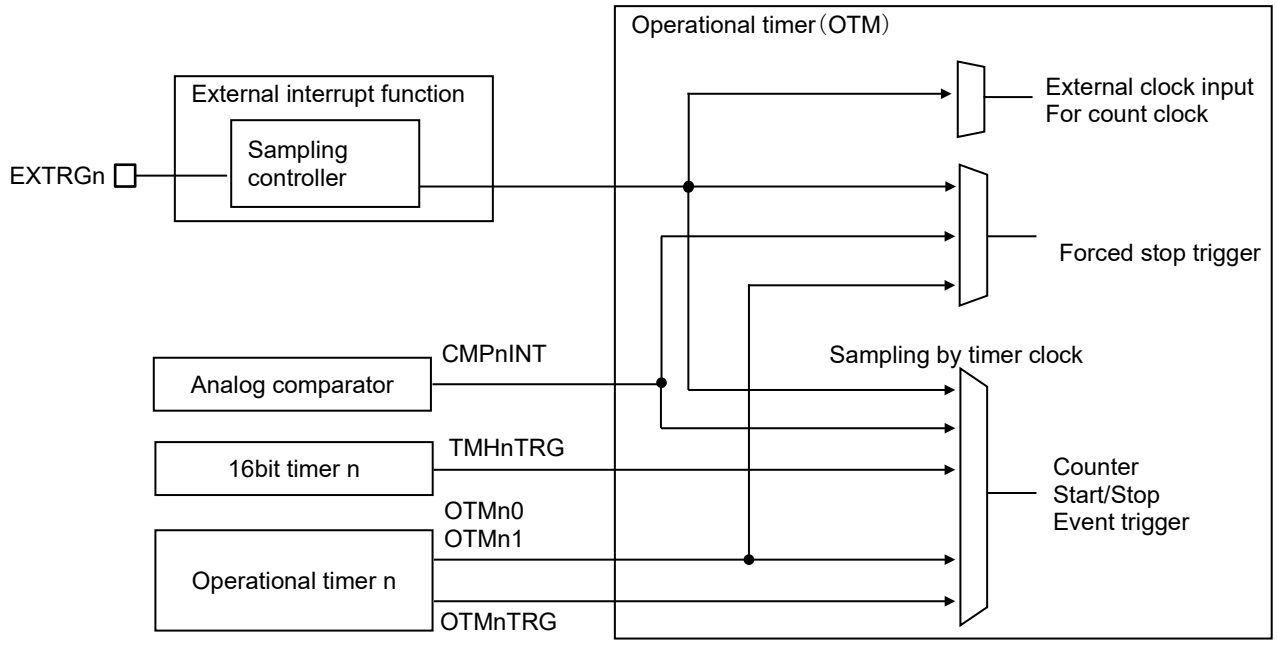


Fig. 9-11 Input Path of Trigger Signal

9.3.6.1 Start/Stop operation by event trigger

The following are the settings for controlling counters by event triggers.
First, set the following items in OTnMOD, OTnCTRG0, and OTnCTRG1 register before controlling the counter.

Setting of OTnMOD register
- After stopping by an event trigger, accept the next counter start or not.
- When OTMn0 output is "H" level when counter is cleared by an event trigger, accept clear counter or not.

Setting of OTnCTRG0, OTnCTRG1 register
- Select whether to enable/disable counter start/stop in an event trigger.
- Select whether to clear counter or not at start/stop due to event triggers.
- Select an event trigger source from EXTRG0 to EXTRG3, CMP0INT to CMP2INT, TMH0TRG, WDTCLK, OTM00 to OTM51, and OTM0TRG to OTM5TRG.
- Setting of accept timing of an event trigger source (edge/level).

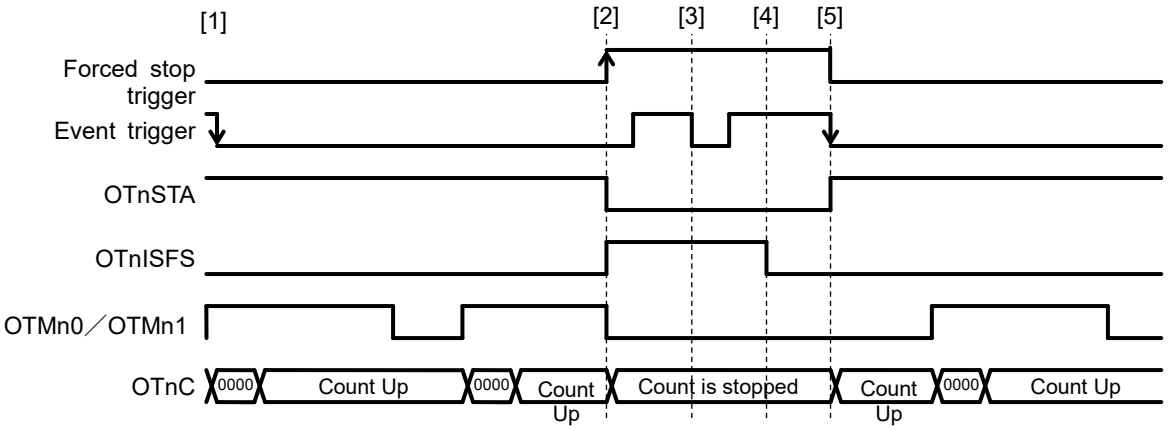
Setting of a timer clock to be used.
Select a timer clock by OTnCLK register.
After the above settings, control the counter with OTCSTR register. Here are the steps:

Step1. Set waiting state of an event trigger
If "1" is written to OTnETG bit, it will be in the waiting state for the event trigger (If the trigger start is set to a level, the counter operation will start as soon as OTnTGEN bit of OTCSTAT register becomes "1" when the corresponding level is reached.).
Step2. Timer count start by software
Write "1" to OTnETG bit and write "1" to OTnSTR bit with the trigger operation enabled, the software starts counting the timer.
Also, if "0" is written to OTnSTP bit of OTCSTP register during counter operation, the software stops counting the timer.

9.3.6.2 Forced stop operation (Counter count stop, Waveform output stop)

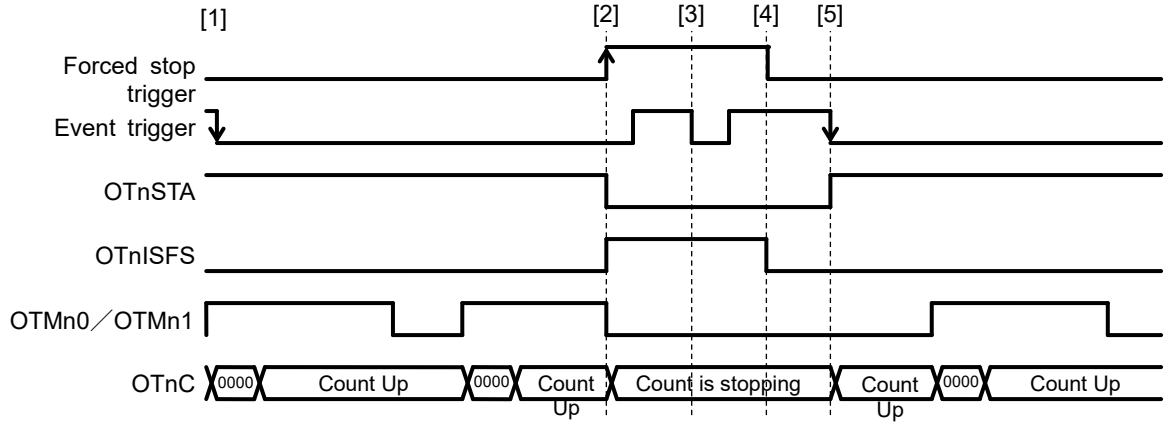
Write "1" to any of OTnFCC0/OTnFCC1, OTnFCN00/OTnFCN10, or OTnFCN01/OTnFCN11 bits of OTnFTRG0/OTnFTRG1 register, the forced stop function is enabled.
When the forced stop function is enabled, a forced stop interrupt occurs when a forced stop trigger is input.
If a forced stop trigger is entered while "1" is being set to OTnFCC0/OTnFCC1 bit of OTnFTRG0/OTnFTRG1 register, the counter stops.
To restart counter operation, write "1" to OTnICFS bit of OTnINTC register to clear the forced stop interrupt status (OTnISFS).
If OTnISFS is "1", the counter will not be restarted. In addition, the forced stop state is also canceled when the counter is cleared by the trigger clear
If the forced stop trigger input is set to "H" level or "L" level, the forced stop interrupt status is automatically cleared after the force stop trigger input is released.
If the forced stop trigger input is set to "H" level or "L" level, the forced stop interrupt status will not be cleared even if "1" is written to OTnICFS bit of OTnINTC register.
If the forced stop trigger is input while "1" is been setting to OTnFCN00/OTnFCN10, OTnFCN01/OTnFCN11 bits of OTnFTRG0/OTnFTRG1 register, the output level of OTMn0/OTMn1 is set to the output level set by OTnSTPOP/N bit of OTnMOD register.

Fig. 9-12-1 and 9-12-2 shows operation timing at forced stop.



[1] Counter operation starts with the event trigger (falling edge).
[2] Counter stop with the forced stop trigger (rising edge). The forced stop interrupt occurs.
[3] The event trigger is disabled because it is being forced stopped.
[4] Clears the forced stop interrupt and becomes operational.
[5] Counter operation restart with the event trigger (falling edge).
(In this example, the counter has not been cleared, so the pulse output is resumed after one period.)

Fig. 9-12-1 Operation timing during forced stop (rising edge)



[1] Counter operation starts with the event trigger (falling edge).
[2] Counter stop with the forced stop trigger (high level). The forced stop interrupt occurs.
[3] The event trigger is disabled because it is being forced stopped.
[4] The forced stop trigger input is released and becomes operational.
[5] Counter operation restart with the event trigger (falling edge).
(In this example, the counter has not been cleared, so the pulse output is resumed after one period.)

Fig. 9-12-2 Operation timing during forced stop (high level)

When the forced stop occurs, the counter is stopped, and OTnISFS bit of OTnINTS register becomes "1" ([2] in Figure 9-12).
When OTnISFS bit is "1", the counter start event trigger is not accepted.
After clearing OTnISFS bit ([4] in Figure 9-12), the counting up restart when the counter start event trigger occurs ([5] in Figure 9-12).
When restart the counting operation by software, OTnISFS bit must be set to "0".
When the forced stop trigger input is set to "H" level or "L" level, OTnISFS bit is automatically cleared after the forced stop input is released (Fig. 9-12-2[4]).

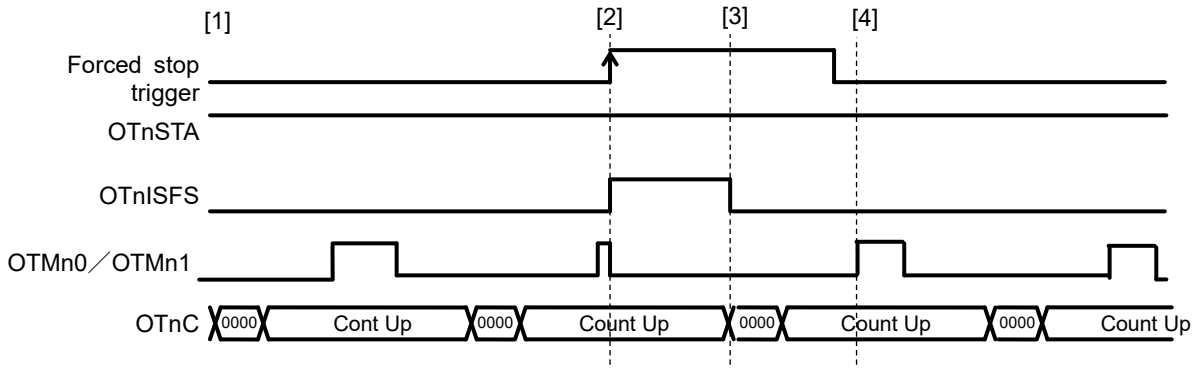
9.3.6.3 Forced stop operation (Counter count continues, Waveform output stop)

When forced to stop, if OTnFCC0/OTnFCC1 bit of OTnFTRG0/OTnFTRG1 register is set to "0", the counter does not stop, fix OTMn0/OTMn1 outputs level to the output level set in OTnSTPO bits of OTnMOD register or hold the current output state by OTnFCN00/OTnFCN10, OTnFCN01/OTnFCN11bits.
The waveform output when stopped are selectable from the following three states.

- Keep in the same output mode
- Fixed at H level (OTnSTPOP/N bit = OTnMOD register =1)
- Fixed at L level (OTnSTPOP/N bit = OTnMOD register =0)

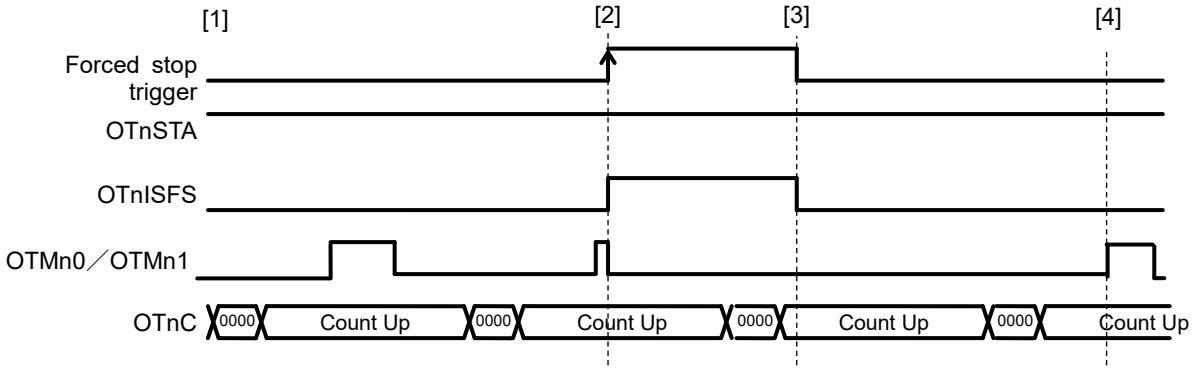
When OTnFCC0/OTnFCC1 bit of OTnFTRG0/OTnFTRG1 registers are set to "0" and the rising/falling edge is selected for the forced stop trigger, the forced stop interrupt status is automatically cleared at the next period when the forced stop trigger is entered.
And also, If OTnFCC0/OTnFCC1 bit of OTnFTRG0/OTnFTRG1 registers are set to "0", and the rising/falling edge is selected for the forced stop trigger, the forced stop interrupt status is automatically cleared at the next period when the force stop trigger is entered.
When the forced stop trigger input is set to "H" level or "L" level, the forced stop interrupt status is automatically cleared after the forced stop is released, and the waveform output restart from the next period after the forced stop is released.
When the forced stop trigger input is set to "H" level or "L" level, the forced stop state is not canceled either when the counter is cleared by the trigger clearing nor when "1" is written to OTnICFS bit of OTnINTC register.

Fig. 9-13-1, 9-13-2 shows the operation timing at the time of forced stop when OTnFCEC bit is set to "1".



[1] Counter operation starts with soft start.
[2] Forced stop trigger (rising edge) input. the forced stop interrupt occurs. The counter operation continues.
[3] Since the next cycle is reached, the forced stop state is cleared and the waveform output is enabled.
[4] After the forced stop state is released, waveform output restart from the next cycle.

Fig. 9-13-1 Operation timing at forced stop (rising edge)



[1] Counter operation starts with soft start.
[2] Forced stop trigger ("H" level) input. the forced stop interrupt occurs. The counter operation continues.
[3] The forced stop trigger input is released and the waveform output is enabled.
[4] After the forced stop state is released, waveform output restart from the next cycle.

Fig. 9-13-2 Operation timing at forced stop ("H" level)

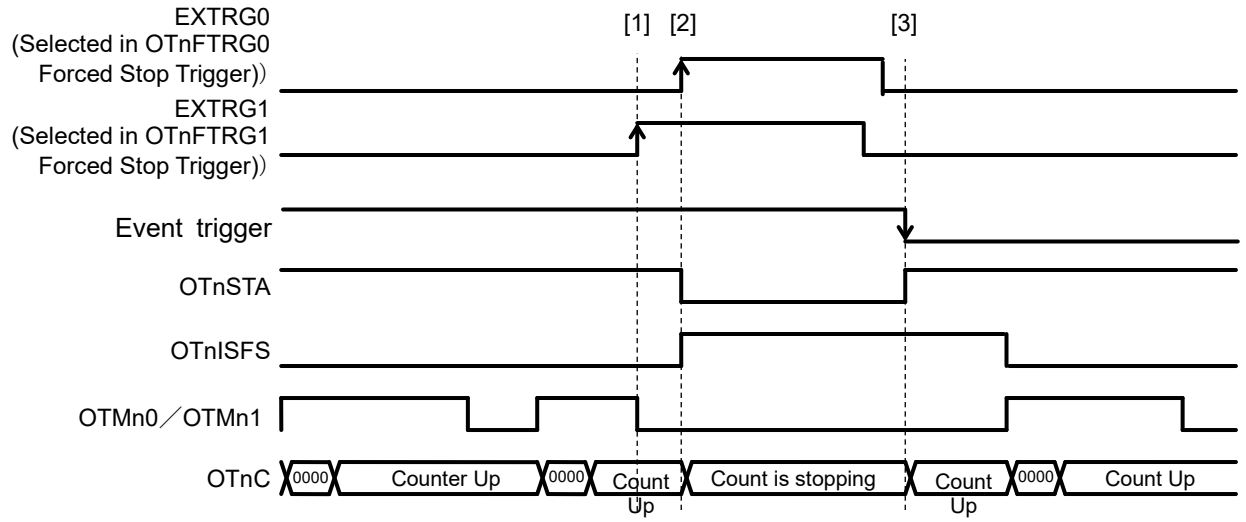
9.3.6.4 Forced stop operation (in case of multiple forced stop triggers are selected)

By using OTnFTRG0/OTnFTRG1 register, it is feasible to use two inputs as forced stop triggers. In addition, it is feasible to set an action for each force stop trigger input.

The following settings are set by OTnFTRG0 register.
Force Stop Trigger : EXTRG0 (OTnEST02 to 00="000")
Force stop event : "H" level (OTnFCE01,00="10")
Stop the counter when forced to stop (OTnFCC0="1")
Output forced stop enabled when forced stop (OTnFCN01=1, OTnFCN00=1)

The following settings are set by OTnFTRG1 register.
Force Stop Trigger :EXTRG1 (OTnEST12 to 10="000")
Force Stop Event :Rising Edge (OTnFCE11,10="00")
Counter continues when forced to stop (OTnFCC1="0")
Output forced stop enabled when forced stop (OTnFCN01=1, OTnFCN00=1)

Fig. 9-14 shows the operation timing of a forced stop when multiple forced stop triggers are selected.



[1] Waveform output is stopped by EXTRG1 forced stop trigger (rising edge). (Counter operation continues)
[2] The counter operation is stopped by EXTRG0 forced stop trigger ("H" level).
[3] Counter operation restart by event trigger input in the forced stop release state by EXTRG0.

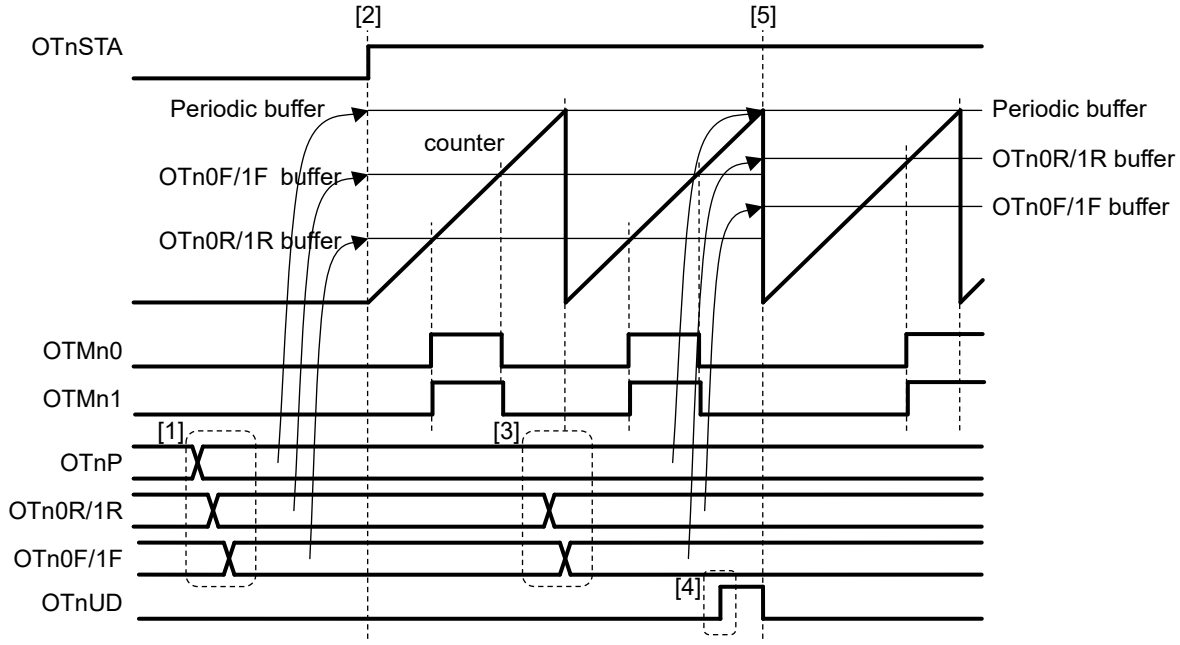
Fig. 9-14 Operation timing at forced stop (Rising edge)

9.3.7 Changing Cycle and Duty during Operation

It is feasible to change the period and the duty according to the setting value of OTnP/OTn0R/OTn0F/OTn1R/OTn1F registers.
The timing of the change differs depending on the counter operation state and the counter value when writing the setting value of OTnP/OTn0R/OTn0F/OTn1R/OTn1F register.

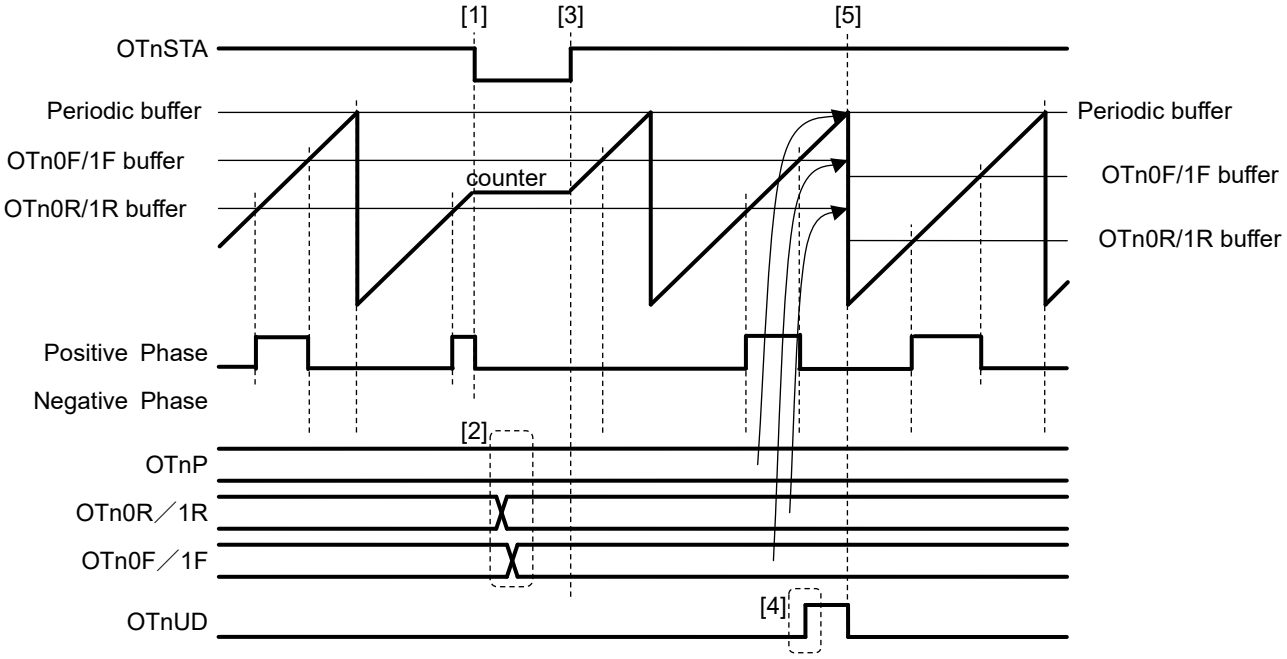
Counter operating status at register setting	Counter value at register setting	Update timing of setting value
stop	0x0000	Update at the counter start
stop	Except 0x0000	Update at the beginning of the period when OTCUDn bit is set to "1" during counter reoperation.
oprating	arbitrary	Update when the beginning of the period when OTCUDn bit is set to "1" during counter operation. Update when OT20R/OT30R/SADR0 is set to OTnP in OTCCCC, the start of the cycle regardless of the setting of OTCUDn bit.
At trigger clear	0x0000	Update when OTCUDn bit is set to "1" during counter operation and the trigger is cleared. Update when OT20R/OT30R/SADR0 is set to OTnP in OTCCCC, the trigger is cleared regardless of OTCUDn bit setting.

Fig. 9-15 shows the operation waveform when the register setting value is updated while the counter is stopped (counter value "0x0000") and when the register setting value is updated while the counter is in operation.
Fig. 9-16 shows the operation waveform when the register setting value is changed while the counter is stopped (other than the counter value "0x0000").



[1] Update OTnP/OTn0R/1R/0F/1F registers while the counter is stopped (the counter value is "0x0000").
[2] Each buffer is updated at the timing of the start of counter operation
[3] Update OTnP/OTn0R/1R/0F/1F registers during counter operation
[4] Write "1" to OTCUDn bit
[5] At the beginning of the period, each buffer is updated and OTCUDn bit is cleared to "0"

Fig. 9-15 When the register setting value is updated during stop (when the counter value is "0x0000") and during operation

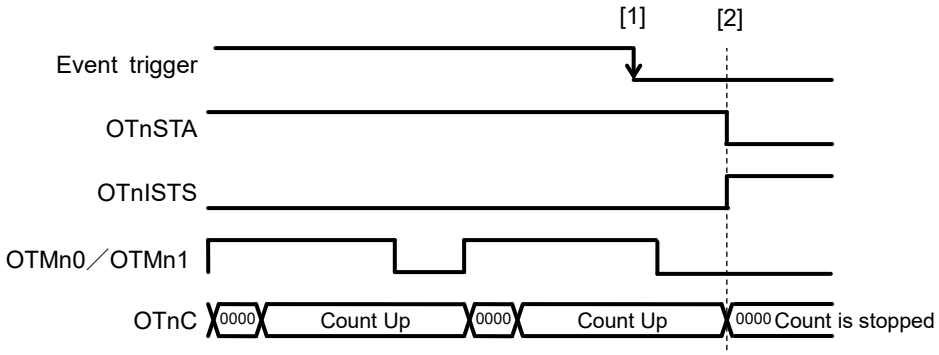


[1] Counter stop
[2] Update OTnP/OTn0R/1R/0F/1F registers while the counter is stopped (the counter value is not "0x0000").
[3] Counter operation restart (each buffer is not updated at this time)
[4] Write "1" to OTCUDn bit
[5] At the beginning of the period, each buffer is updated and OTCUDn bit is cleared to "1".

Fig. 9-16 when the register setting value is changed during stop (other than the counter value is "0x0000") and during operation

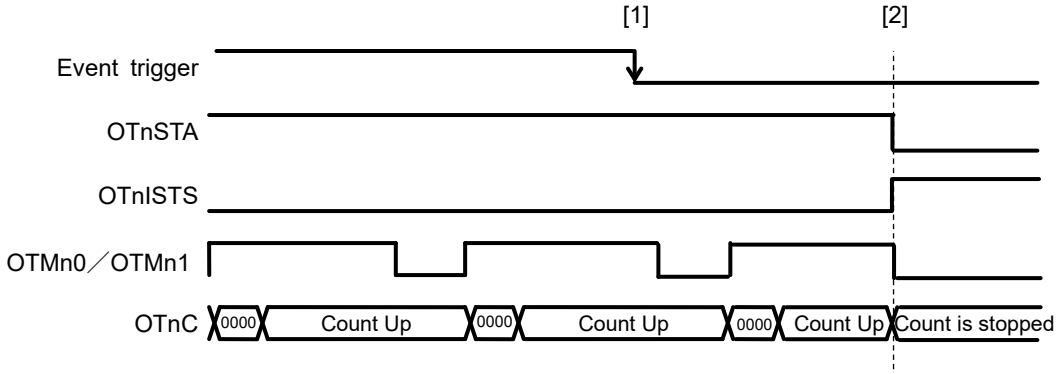
9.3.8 Cycle stop function

Cycle stop function is a function that counts to the expiration of the period (OTnP) + set value (OTnSCLR) before stopping the counter when a request to stop is received by software or a trigger event during timer operation.
Fig. 9-17-1, 9-17-2 shows the operation timing when a stop due to a trigger event is entered when OTnSTSYN bit is set to "1".



[1] Counter stop request with event trigger (falling edge). The counter does not stop at this moment.
[2] Counter stop at the end of the cycle. Trigger counter stop interrupt (OTnISTS bit of OTnINTS) occur by event trigger.

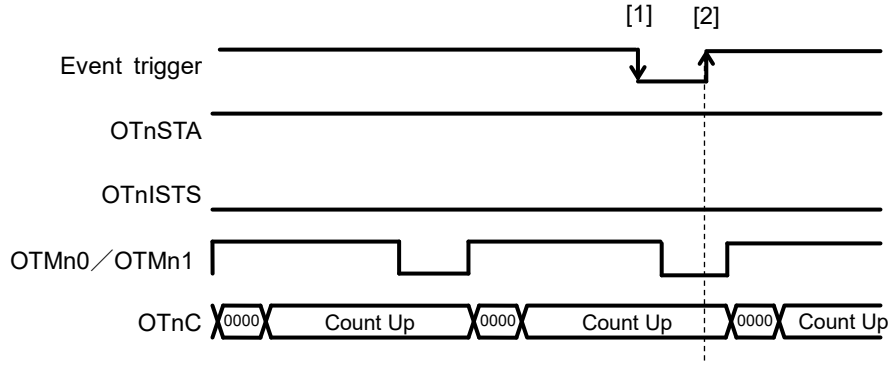
Fig. 9-17-1 Operation timing when trigger event is stopped（OTnSTSYN="1"/OTnDLT7 to 0="0x00"）.



[1] Counter stop input with event trigger (falling edge). The counter does not stop at this moment.
[2] After counting the interval set by OTnDLY7 to 0 from the end of the cycle, stop the counter. The interrupt occurs due to input stop by event trigger.

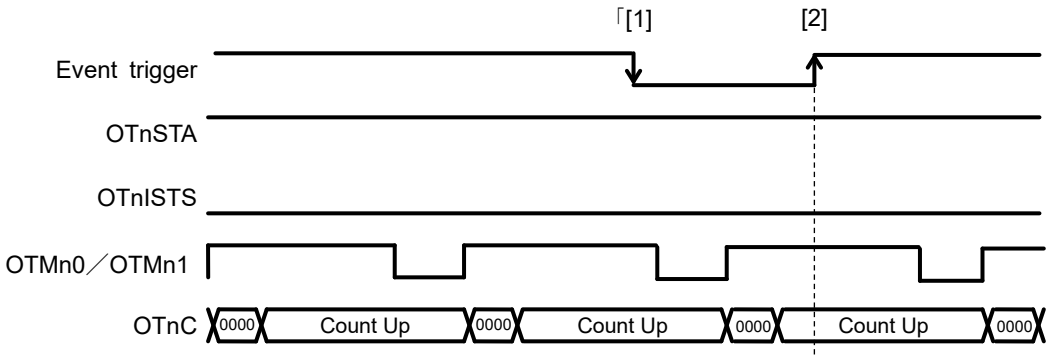
Fig. 9-17-2 Operation timing when trigger event is stopped (OTnSTSYN="1"/OTnDLT7 to 0≠"0x00")

If a request to stop is received when the cycle stop function is enabled, and a request to start the counter is received by the software or trigger while counting to the expiration of the period (OTnP) + the set value (OTnSCLR), the request to stop is canceled.
Fig. 9-18-1 and Fig. 9-18-2 shows the operation timing when a stop due to a trigger event is entered when OTnSTSYN bit is set to "1" and when a counter start by a trigger event is entered before the counter stops.



[1] Counter stop input by the event trigger (falling edge). The counter does not stop at this point
[2] Counter start input by the event trigger (rising edge) before counter stop. Counter operation continues thereafter.

Fig, 9-18-1 Operation timing when trigger event stop is released (OTnSTSYN="1"/OTnDLT7 to 0="0x00")



[1] Counter stop input by the event trigger (falling edge). The counter does not stop at this point
[2] Counter start input by the event trigger (waiting for the interval set by OTnDLY7 to 0) before counter stop. Counter operation continues thereafter. Counter operation continues thereafter.

Fig. 9-18-2 Operation timing when trigger event stop is released (OTnSTSYN≠"1"/OTnDLT7 to 0="0x00")

9.3.9 Interrupt

Writing "1" to the corresponding bit (OTnIE*) of OTnINTE register causes each interrupt request to be enabled.
Note that permission of the forced stop interrupt is not available. If the forced stop function is enabled, the interrupt is also enabled.
For the source which caused the interrupt status to become "1", write "1" to each interrupt status clear bit (OTnIC*) to clear each interrupt status bit (OTnIS*).
If using an interrupt, clear each interrupt status bit (OTnIS*) at the end of the interrupt routine.

Name	Mode	Status	Method of clear
Periodic interrupts	All mode	OTnISP bit	Write "1" to OTnICP bit
0 Rise Point interrupt	PWM	OTnIS0R bit	Write "1" to OTnIC0R bit
	CAPTURE	OTnIS0R bit	Write "1" to OTnIC0R bit or Read out OTn0R register
0 Fall Point B interrupt	PWM	OTnIS0F bit	Write "1" to OTnIC0F bit
	CAPTURE	OTnIS0F bit	Write "1" to OTnIC0F bit or Read out OTn0F register
1 Rise Point interrupt	PWM	OTnIS1R bit	Write "1" to OTnIC1R bit
	CAPTURE	Unused	-
1 Fall Point B interrupt	PWM	OTnIS1F bit	Write "1" to OTnIC1F bit
	CAPTURE	Unused	-
Trigger stop interrupt	All mode	OTnISTS bit	Write "1" to OTnICTS bit
Trigger start interrupt	All mode	OTnISTR bit	Write "1" to OTnICTR bit
Forced stop interrupt	All mode	OTnISFS bit	Write "1" to OTnICFS bit

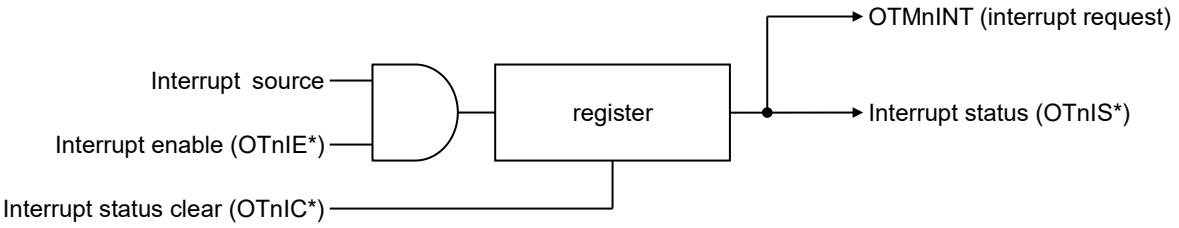


Fig. 9-19 Interrupt Control Signals

9.3.10 Clock Mutual Monitoring Function

It is monitorable whether the low-speed clock (low-speed RC oscillator circuit) and the high-speed clock (PLL oscillator circuit) oscillate normally using 16-bit timer and operational timer
Fig. 9-20 shows an example of oscillation monitoring operation of a high-speed clock (PLL oscillator circuit) using a 16-bit timer 0 and an operational timer 0.

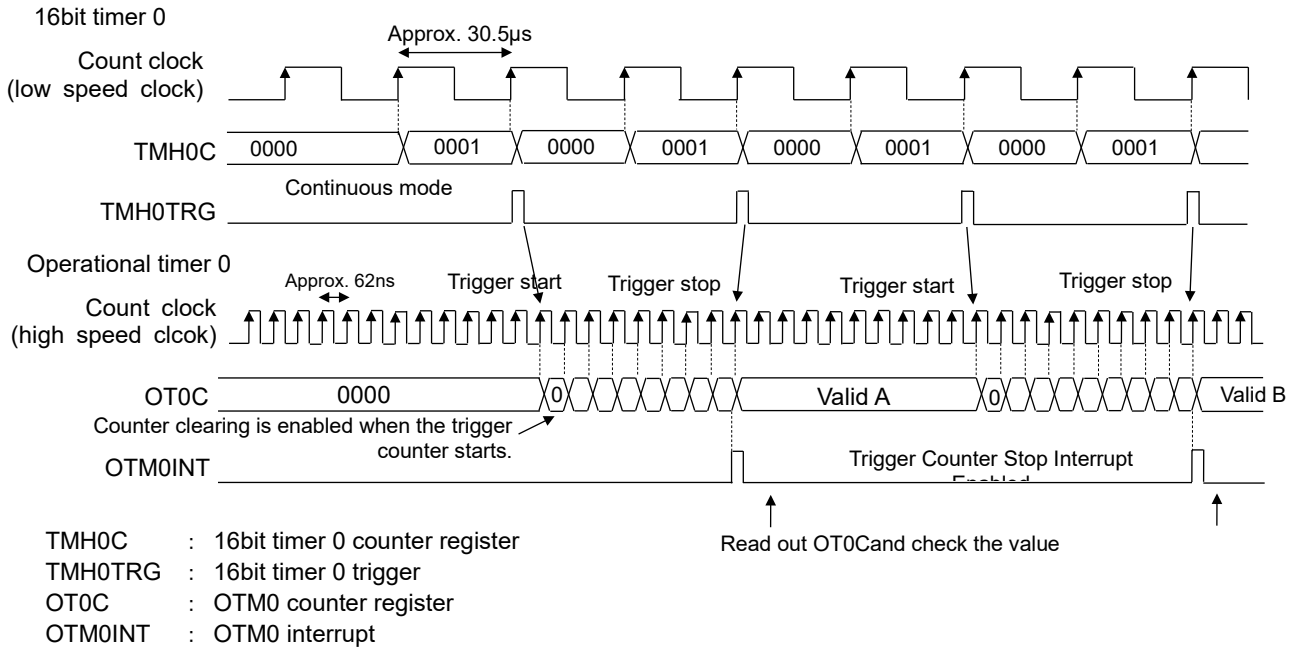


Fig. 9-20 Example of mutual monitoring of low-speed clock and high-speed clock

Fig. 9-21 shows the settings for the example operation shown in Fig. 9-20.

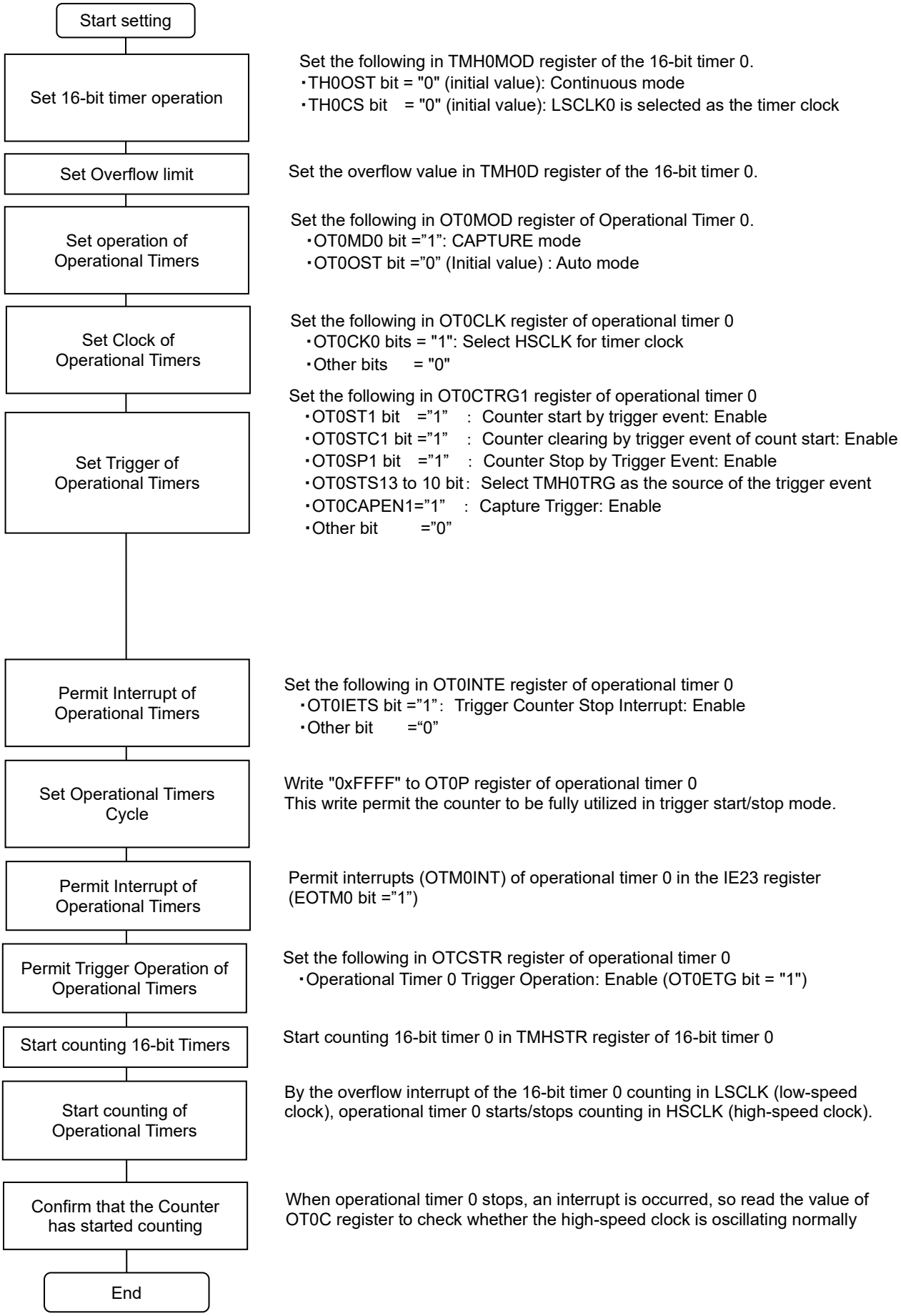


Fig. 9-21 Setting of an oscillation monitoring operation example of a high-speed clock (PLL oscillator circuit)

Chapter 10 Watchdog Timer

10. Watchdog Timer

10.1 General Description

The watchdog timer (WDT) is equipped with the following functions and can detect the runaway state of program or the undefined state of the CPU by generating an interrupt or reset when an abnormality occurs.

- If the counter is not cleared for more than a certain time period in program operation and overflows, the WDT interrupt is generated in the first overflow and the WDT reset in the second overflow (if the window function is disabled).
- If the counter is not cleared for more than a certain time period in program operation and overflow occurs, the WDT reset is generated in the first overflow (if the window function is enabled).
- If the counter is cleared in the unexpected time period, the WDT invalid clear reset is generated (if the window function is enabled).

The window function refers to the function through which "the time period during which WDT counter clear is enabled" = "the time period during which the window is opened" and
"the time period in which WDT counter clear is disabled" = "the time period in which the window is closed" can be set.

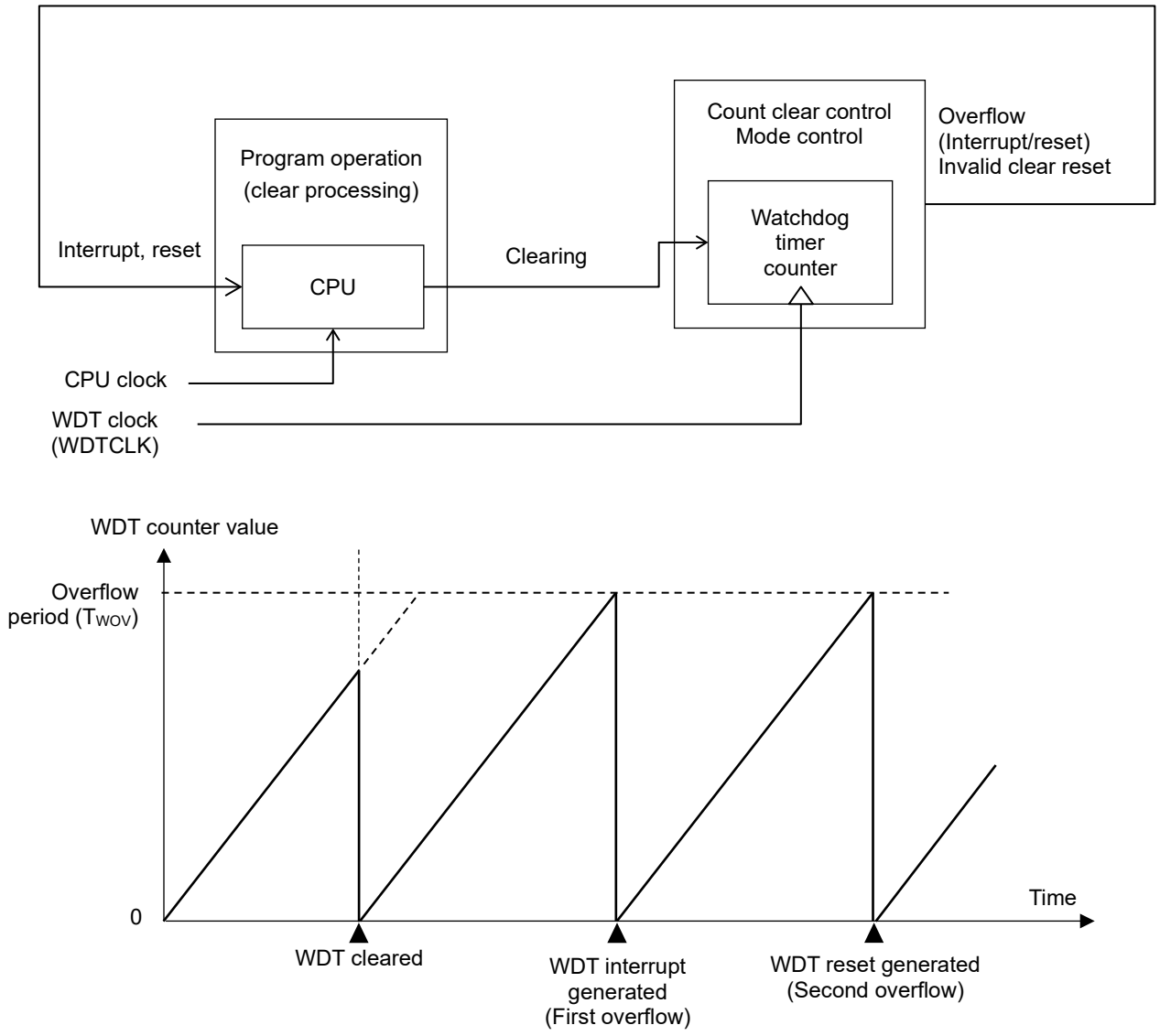


Figure 10-1 Watchdog Timer Overview (With the Window Function Disabled)

10.1.1 Features

- Eight types of overflow periods can be selected (7.8ms, 15.6ms, 31.3ms, 62.5ms, 125ms, 500ms, 2s, or 8s)
- Two operation modes:
  - Window function disabled mode
    The WDT counter can always be cleared. The WDT interrupt is generated when the first counter overflow occurs, and the WDT reset is generated when the second counter overflow occurs.
  - Window function enabled mode
    The periods during which WDT counter clear is enabled and disabled respectively can be set. The WDT reset is generated when the first counter overflow occurs, and the WDT invalid clear reset is generated when the counter is cleared in the period during which WDT counter clear is disabled.

Table 10-1 Watchdog Timer Operation Modes

Mode	overflow		WDT invalid clear reset
	First	Second	
Window function disabled mode	Interrupt	Reset	-
Window function enabled mode	Reset	-	Reset

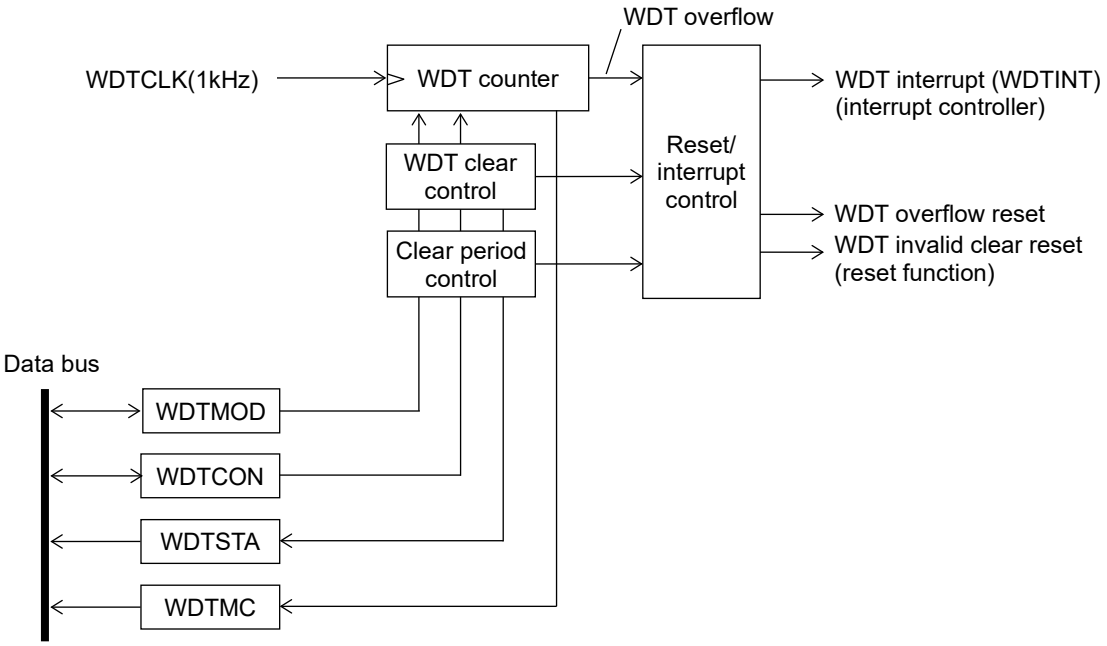
- The following items can be selected by the code option. See the Chapter 30 "Code Option" for details of the code option.
  - Enabling/disabling the WDT timer operation
  - Enabling/disabling the WDT timer operation in HALT mode

[Note]

- **Not all the abnormal operations are detectable by the watchdog timer. Even if the CPU is runaway, the watchdog timer is undetectable to the abnormality in the operation state in which the WDT counter is cleared. As a fail-safe, it is recommended that the WDT counter is cleared at one place in the main loop of the program.**

10.1.2 Configuration

The following diagram shows the configuration of the watchdog timer.



WDTCON : Watchdog timer control register
WDTMOD : Watchdog timer mode register
WDTMC : Watchdog timer counter register
WDTSTA : Watchdog timer status register

Figure 10-2 Configuration of Watchdog Timer

10.2 Description of Registers

10.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF010	Watchdog timer control register	WDTCON	-	R/W	8	0x00
0xF011	Reserved	-	-	-	-	-
0xF012	Watchdog timer mode register	WDTMOD	-	R/W	8	0x06
0xF013	Reserved	-	-	-	-	-
0xF014	Watchdog timer counter register	WDTMCL	WDTMC	R	8/16	0x00
0xF015		WDTMCH		R	8	0x00
0xF016	Watchdog timer status register	WDTSTA	-	R	8	0x01
0xF017	Reserved	-	-	-	-	-

10.2.2 Watchdog timer control register (WDTCON)

WDTCON is a SFR to clear the WDT counter.

Address: 0xF010 (WDTCON)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTCON							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	WDP/d0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	d7 to d0	WDT counter is cleared by writing "0x5A" with WDP bit set to "0", then writing "0xA5" with WDP bit set to "1". In the window mode, WDT invalid clear reset is generated if WDT counter is cleared in the period during which WDT clear is disabled.
0	WDP	Read the value of WDT bit internal pointer. WDP bit is reset to "0" when the system is reset as well as when WDT counter overflows. It is reversed when writing data to WDTCON register regardless of the data written.

[Note]

- **In WDT interrupt routine (when the interrupt level (ELEVEL) of the CPU program status word (PSW) is "2"), WDT counter is unable to get cleared.**

10.2.3 Watchdog Timer Mode Register (WDTMOD)

WDTMOD is a SFR to set the overflow period and the clear enabled period of the WDT counter.

Address: 0xF012 (WDTMOD)
Access: R/W
Access size: 8 bit
Initial value: 0x06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	WOVF1	WOVF0	-	WDT2	WDT1	WDT0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit No.	Bit symbol name	Description
7, 6	-	Reserved bits
5, 4	WOVF1, WOVF0	Set the mode of WDT. 00: Window function disabled (initial value) 01: Window function enabled mode 1 (the clear enabled period is approximately 75% of the overflow period) 10: Window function enabled mode 2 (the clear enabled period is approximately 50% of the overflow period) 11: Setting disabled (setting of window function enabled mode 2) If the overflow period of the WDT counter is set to 62.5 ms or less in WDT2 to 0 bits, the window function is disabled regardless of setting values of WOVF1 and WOVF0 bits.
3	-	Reserved bits
2 to 0	WDT2 to WDT0	Set the overflow period (T_{WOV}) of the WDT counter. 000: Approx. 7.8ms 001: Approx. 15.6ms 010: Approx. 31.3ms 011: Approx. 62.5ms 100: Approx. 125ms 101: Approx. 500ms 110: Approx. 2s (initial value) 111: Approx. 8s where frequency of WDTCLK is 1.024kHz.

[Note]
- **See the data sheet for frequency accuracy of RC1K.**

10.2.4 Watchdog Timer Counter Register (WDTMC)

WDTMC is a SFR to read the WDT counter value.

Address: 0xF014 (WDTMCL/WDTMC), 0xF015 (WDTMCH)
Access: R
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	WDTMC															
Byte	WDTMCH								WDTMCL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	d15 to d0	Read the WDT counter value. The normal counting operation of the WDT counter can be confirmed If values periodically read from the WDT counter vary.

10.2.5 Watchdog Timer Status Register (WDTSTA)

WDTSTA is a read-only special function register (SFR) to indicate the WDT counter clearing state.

Address: 0xF016 (WDTSTA)
Access: R
Access size: 8 bit
Initial value: 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTSTA							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WDTWIN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	WDTWIN	Indicate the status of enabling/disabling WDT counter clearing. It is always “1” when a window function is disabled 0: Clearing is disabled. 1: Clearing is enabled (initial value).

10.3 Description of Operation

The WDT counter starts counting at the rising edge of the WDT counter operation clock (WDTCLK) when the system reset is released. (Enable/disable watchdog timer is set in the code options)

The WDT counter can be cleared by writing "0x5A" to WDTCON register with the WDP bit set to "0", then writing "0xA5" to WDTCON register with the WDP bit set to "1" while WDT counter clearing is enabled.
The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed every time data is written to WDTCON register.

Two types of use are available: window function disabled mode and window function enabled mode.

- Window function disabled mode
  The WDT counter can always be cleared. The WDT interrupt is generated when the counter overflows for the first time, and the WDT reset is generated when the counter overflows a second time.
- Window function enabled mode
  The periods during which WDT counter clear is enabled and disabled respectively can be set. The WDT reset is generated when the counter overflows for the first time, and the WDT invalid clear reset is generated when the counter is cleared in the period during which WDT counter clear is disabled.

Table 10-2 Watchdog Timer Operation Modes

Mode	Overflow		WDT invalid clear reset
	First	Second	
Window function disabled mode	Interrupt	Reset	-
Window function enabled mode	Reset	-	Reset

The WDT counter overflow period (T_{WOV}) and the WDT counter clear enabled period (T_{WCL}) can be selected through the WDTMOD register.

The following items can be selected with the code option. See Chapter 30 "Code Option" for details on how to set the code option.

- Enabling/disabling the WDT timer operation
- Enabling/disabling the WDT timer operation in the HALT mode

10.3.1 How to Clear WDT Counter

The WDT counter can be cleared by writing "0x5A" to the WDTCON register with the WDP bit set to "0", then writing "0xA5" to the WDTCON register with the WDP bit set to "1" while WDT counter clearing is enabled.
The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed every time data is written to the WDTCON register.

The following diagram shows the WDT counter clearing timing chart.

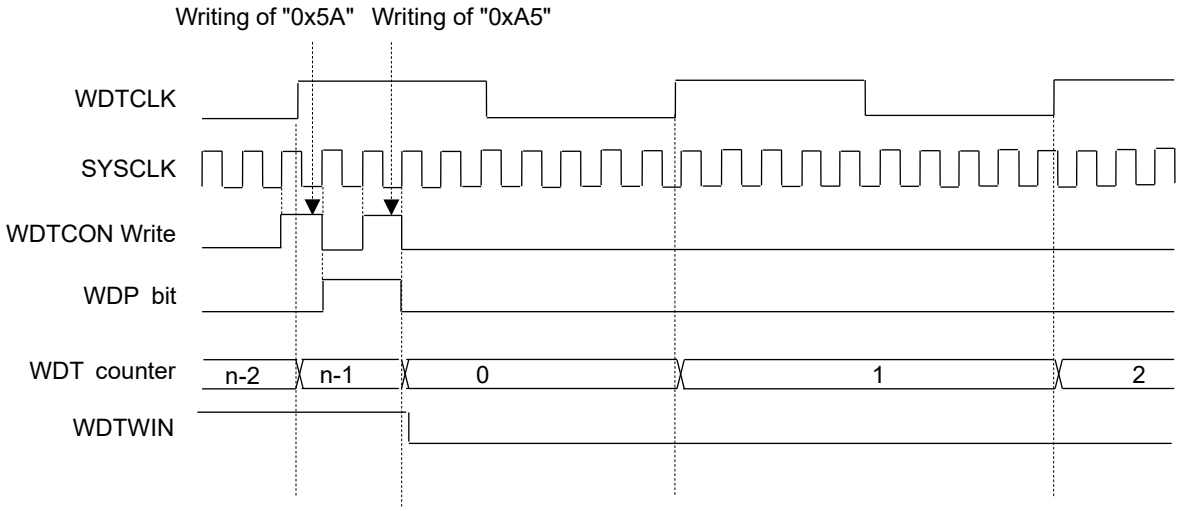


Figure 10-3 WDT Counter Clearing Timing Chart

The following description shows a sample program script of the watchdog timer.

```
void wdt_clear( void )
  {
    unsigned char pswval;

    if(WDTCLR1 == 1) {                  // Checking presence of pending clearing process
      return;
    };
    if(WDTCLR2== 1) {                   // Checking whether clearing process is pending or completed
      return;
    };

    pswval = s_drvcommon_getPSW();      // Saving PSW
    _DI();                              // Interrupt disabled (clearing MIE bit)
    do {
         WDTCON = 0x5A;                 // WDT counter clearing
      } while (WDP != 1 );
    WDTCON = 0xA5;
    if ((pswval & 0x08) != 0) {         // Confirming MIE bit
      _EI();                            // Interrupt enabled (setting MIE bit)
    }

    static unsigned char s_drvcommon_getPSW( void ){
      #pragma asm
      mov r0,psw
      rt
      #pragma endasm
    }
```

Figure 10-4 Sample Program Script of Watchdog Timer

[Note]

- **In the STOP mode, the WDT timer is stopped.**

10.3.2 Window Function Disabled Mode

In the window function disabled mode, if the WDT counter is not available to clear within the WDT counter overflow period (T_{WOV}) and the counter overflows for the first time, a WDT interrupt is generated. If the WDT counter is not cleared even by the software processing after the WDT interrupt, and overflows again, a WDT reset occurs.
The WDTR bit of RSTAT register is set to "1" when the WDT reset occurs, and the state on the LSI is transferred to the system reset mode. See Chapter 3 "Reset Function" for details of RSTAT register.

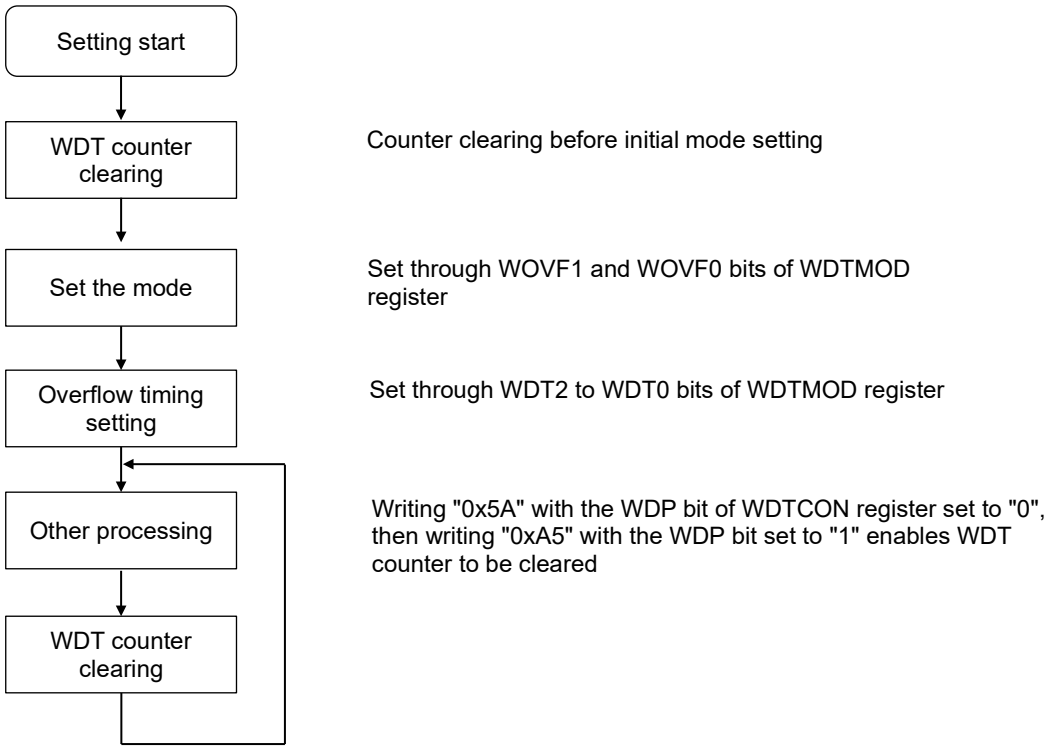


Figure 10-5 Procedure to Use WDT (in Window Function Disabled Mode)

The following figure shows an operation timing overview of the window function disabled mode.

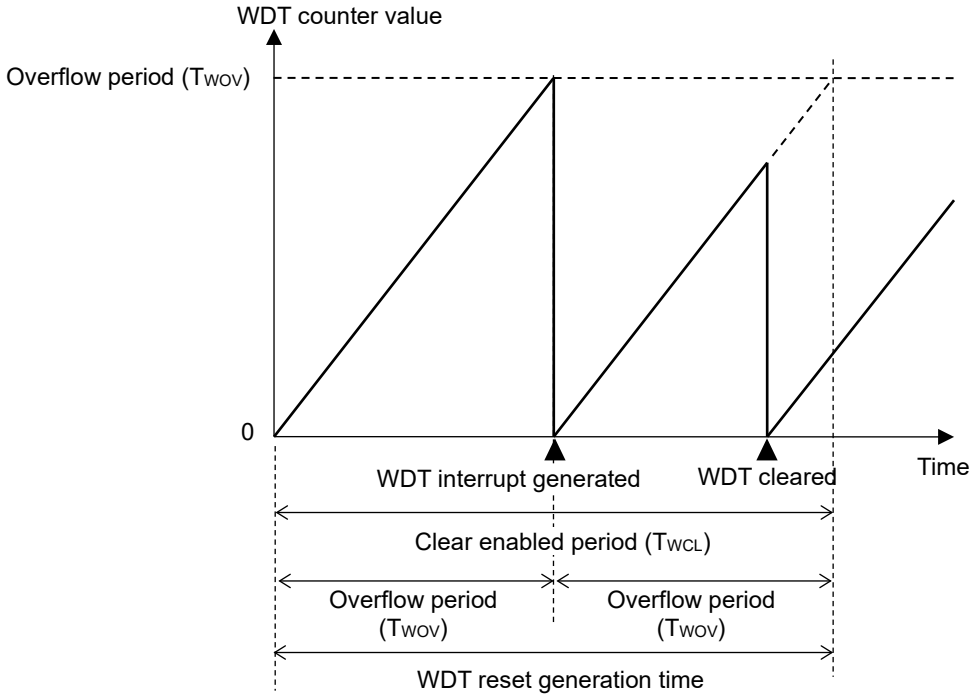


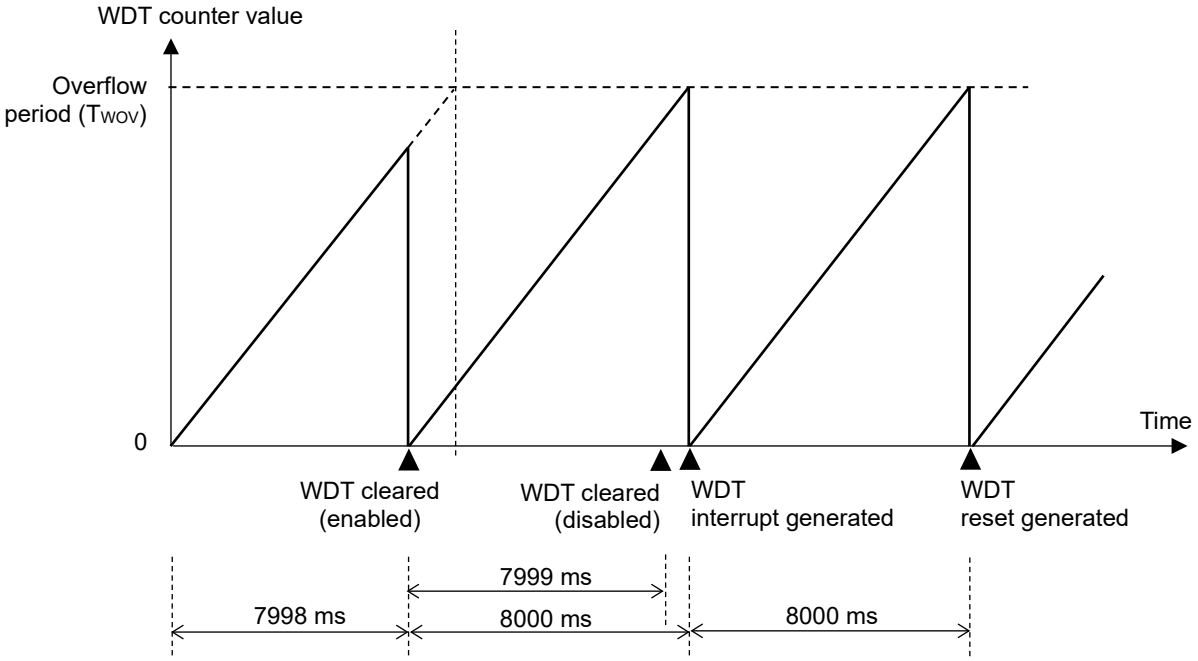
Figure 10-6 Overview of Operation Timing in Window Function Disabled Mode

The following table shows the WDT counter clear enabled period in the window function disabled mode.

Table 10-3 WDT Counter Clear Enabled Period in Window Function Disabled Mode

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT counter clear enabled period (T_{WCL})*1
0	0	0	7.8 ms	15.6 ms	≈ Overflow period
0	0	1	15.6 ms	31.3 ms	≈ Overflow period
0	1	0	31.3 ms	62.5 ms	≈ Overflow period
0	1	1	62.5 ms	125 ms	≈ Overflow period
1	0	0	125 ms	250 ms	≈ Overflow period
1	0	1	500 ms	1000 ms	≈ Overflow period
1	1	0	2000 ms	4000 ms	≈ Overflow period
1	1	1	8000 ms	16000 ms	≈ Overflow period

*1: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.



Design the WDT clear timing with time to spare.

Figure 10-7 Example of Operation Timing in Window Function Disabled mode (When Overflow Period=8000 ms)

The following figure shows details of operation timing in the window function disabled mode.

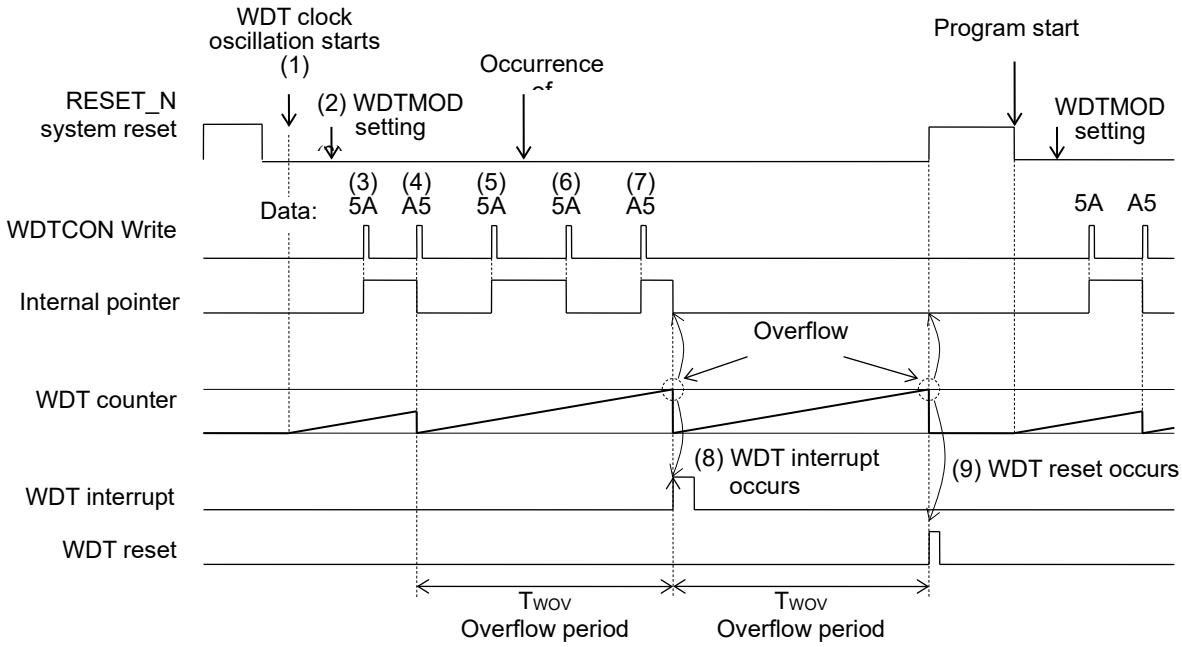


Figure 10-8 Details of Operation Timing in Window Function Disabled Mode

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDT counter overflow period (TWOV) is set to the WDTMOD register.
(3) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1→0)
(5) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)
(6) When "0x5A" is written to the WDTCON register after an abnormality occurred, it is not accepted because the internal pointer WDP is "1". (Internal pointer WDP: 1 →0)
(7) Although "0xA5" is written to the WDTCON register, the WDT counter is not cleared because the internal pointer WDP is "0" and writing of "0x5A" is not accepted in (6). (Internal pointer WDP: 0 →1)
(8) The WDT counter overflows and a WDT interrupt request is generated. (Internal pointer WDP: 1 →0)
Following cleared due to the overflow, the WDT counter continues counting up.
(9) If the WDT counter is not cleared even by the software processing after the WDT interrupt and it overflows again, a WDT reset occurs and the shift to the system reset mode takes place.

10.3.3 Window Function Enabled Mode

In the window function enabled mode, if the WDT counter is not available to clear within the WDT clear enabled period and the counter overflows first time, the WDT overflow reset is generated.
In addition, if the WDT counter is cleared in the period the counter clear is not enabled, the WDT invalid clear reset is generated.

WDTR bit of RSTAT register is set to "1" when the WDT overflow reset occurs, and the state on the LSI is transferred to the system reset mode.
WDTWR bit of RSTAT register is set to "1" when the WDT invalid clear reset occurs, and the state on the LSI is transferred to the system reset mode. See Chapter 3 "Reset Function" for details of RSTAT register.

In the window function enabled mode, two types of modes can be selected through WDTMOD register:

- Window function enabled mode 1 (the clear enabled period is approximately 75% of the overflow period)
- Window function enabled mode 2 (the clear enabled period is approximately 50% of the overflow period)

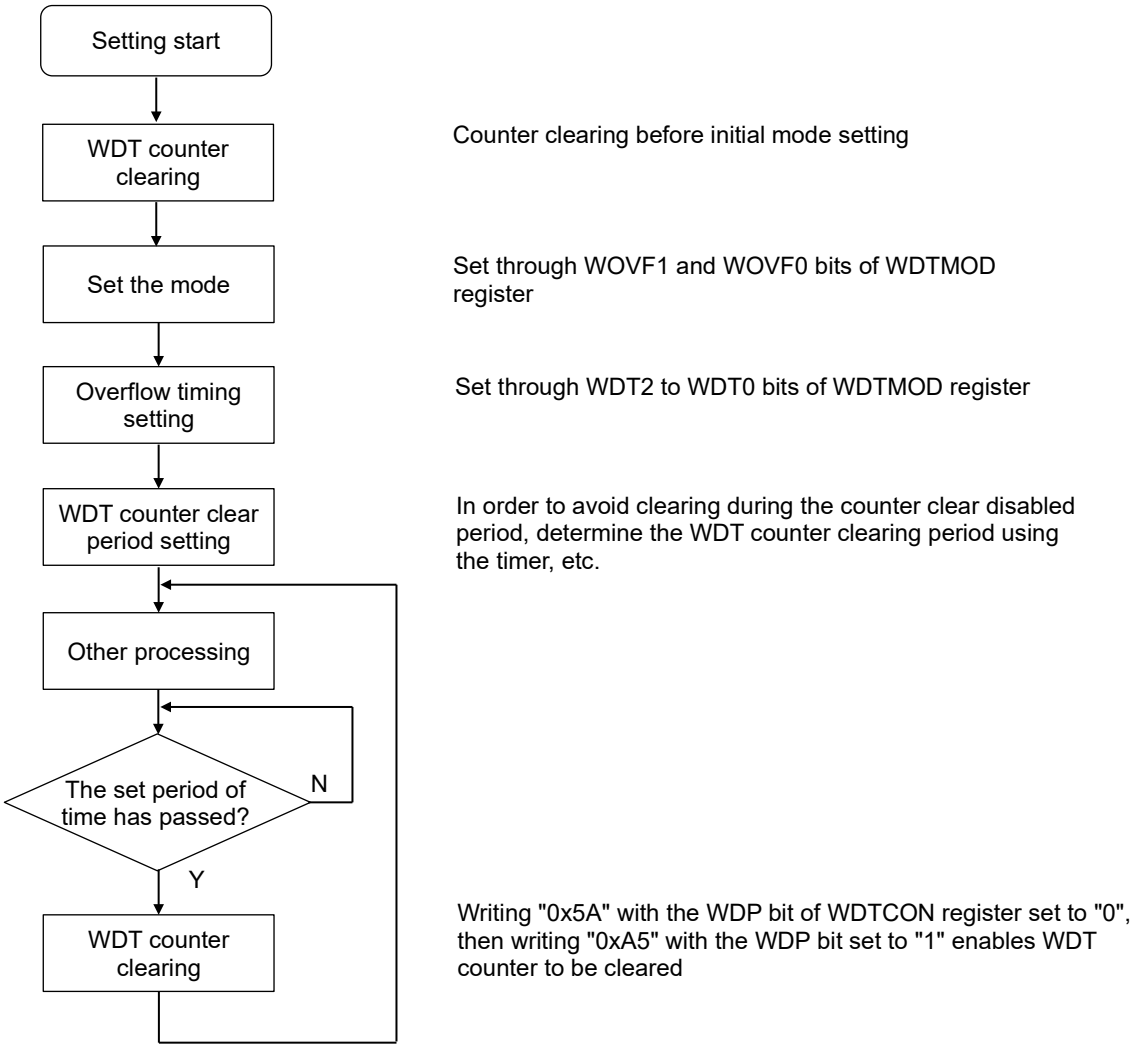


Figure 10-9 Procedure to Use WDT (in Window Function Enabled Mode)

Overviews of the operation of each mode are shown below.

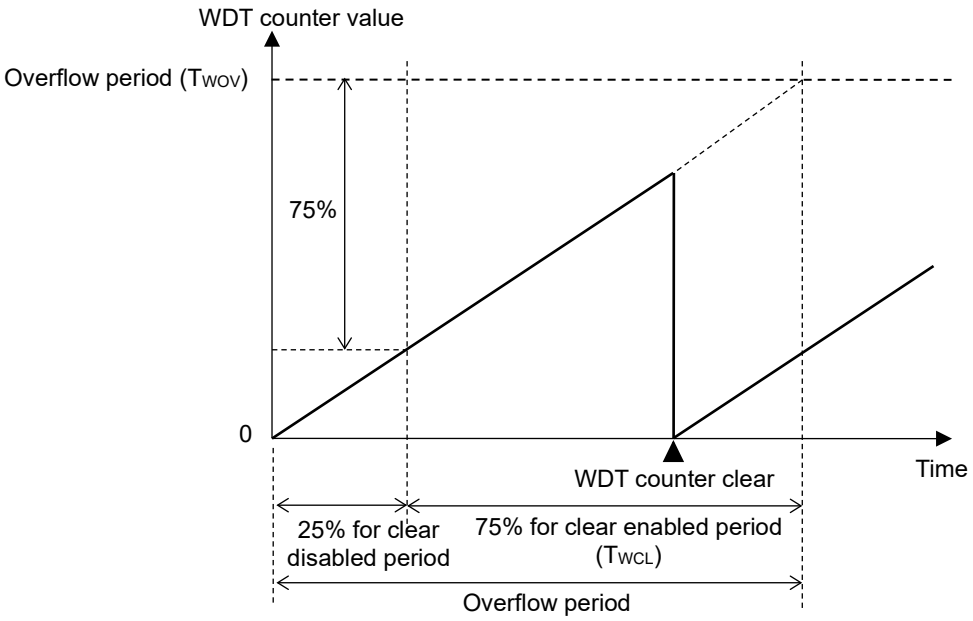


Figure 10-10 Window Function Enabled Mode 1 Operation Overview

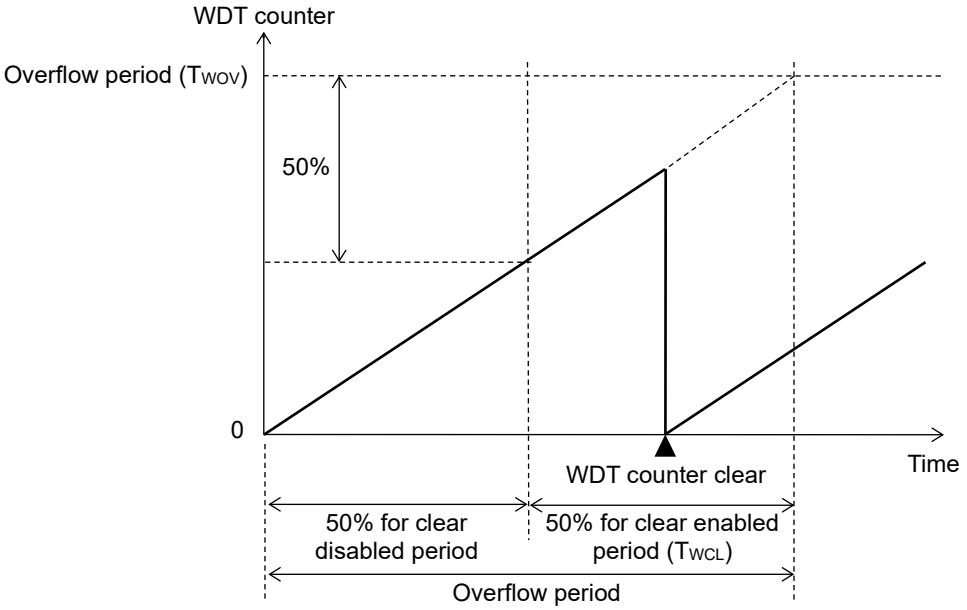


Figure 10-11 Window Function Enabled Mode 2 Operation Overview

The following table shows WDT counter clear enabled periods.
If the overflow period of the WDT counter is set to 62.5 ms or less in WDT2 to 0 bits, the window function is disabled regardless of setting values of WOVF1 and WOVF0 bits.

Table 10-4 WDT Clear Enabled Period in Window Function Enabled Mode 1

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT clear enabled period (T_{WCL})*1
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 75% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 75% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 75% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 75% of overflow period

*1: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.

Table 10-5 WDT Counter Clear Enabled Period in Window Function Enabled Mode 2

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT clear enabled period (T_{WCL})*1*2
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 50% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 50% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 50% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 50% of overflow period

*1: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.

[Note]

- **When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.**

The following figure shows details of operation timing in the window function enabled mode.

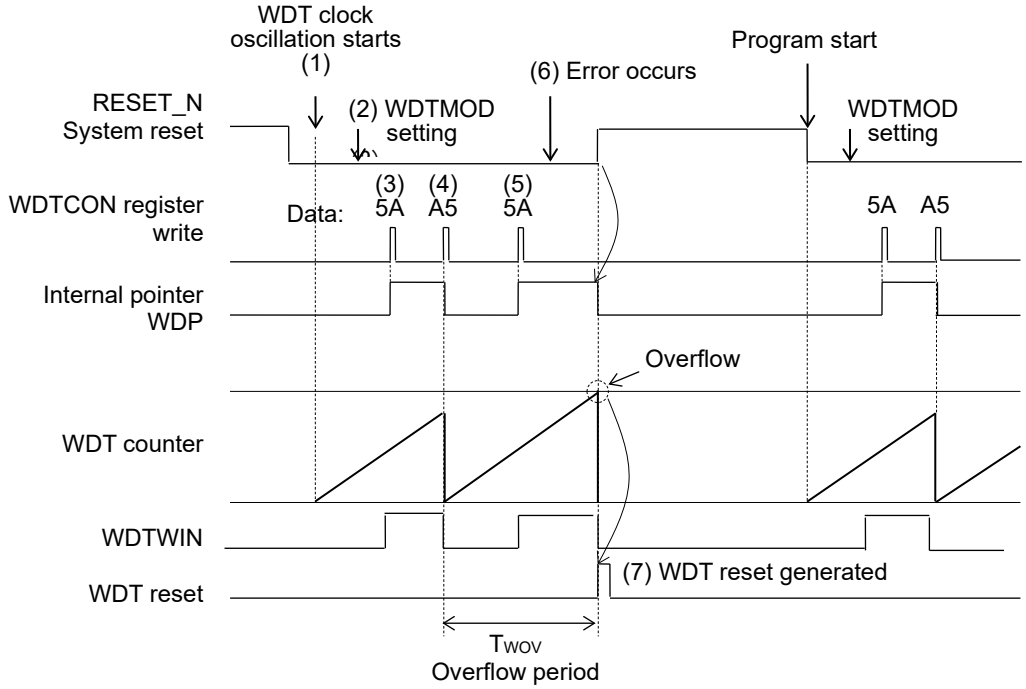


Figure 10-12 Details of Operation Timing in Window Function Enabled Mode

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (T_{WOV}) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: “0” → “1”)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: “1” → “0”)
(5) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: “0” → “1”)
(6) Occurrence of abnormality
(7) The WDT counter overflows and a WDT reset occurs. (Internal pointer WDP: “1” → “0”)

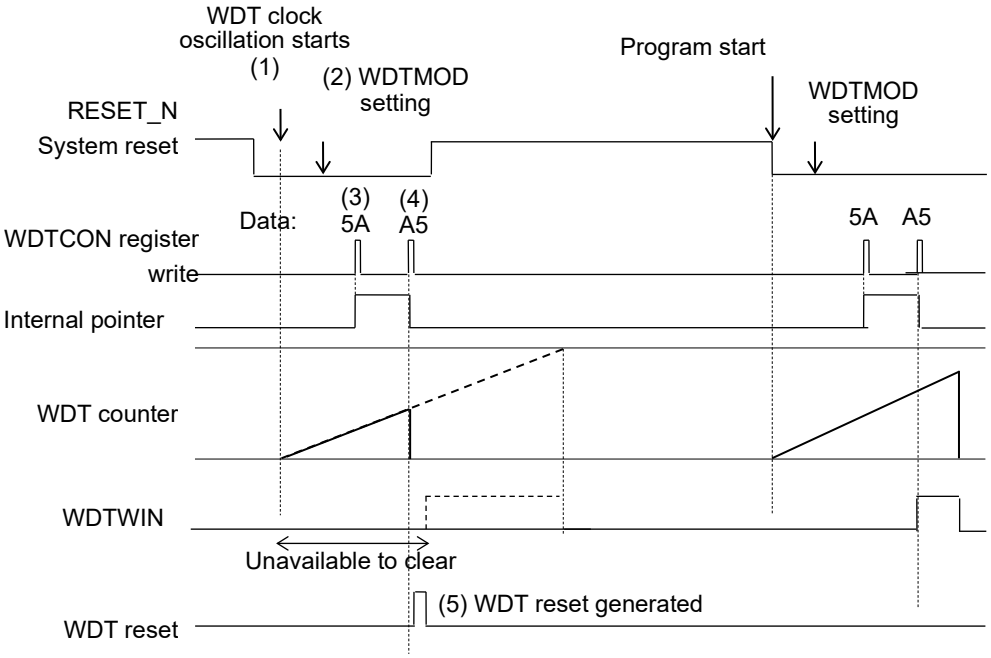


Figure 10-13 WDT invalid clear reset

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (T_{WOV}) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON. (Internal pointer WDP: “0” → “1”)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: “1” → “0”)
(5) WDT invalid clear reset is occurred by clear processing during the WDT clear disabled period.

[Note]

- **In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.**

Chapter 13 I^2C Bus

13. I²C Bus

13.1 General Description

This LSI has one channel of I²C bus unit that supports both master and slave function.
I²C bus unit is that either of master or slave is selected to use and both functions of master and slave are unworkable at the same time.

13.1.1 Features

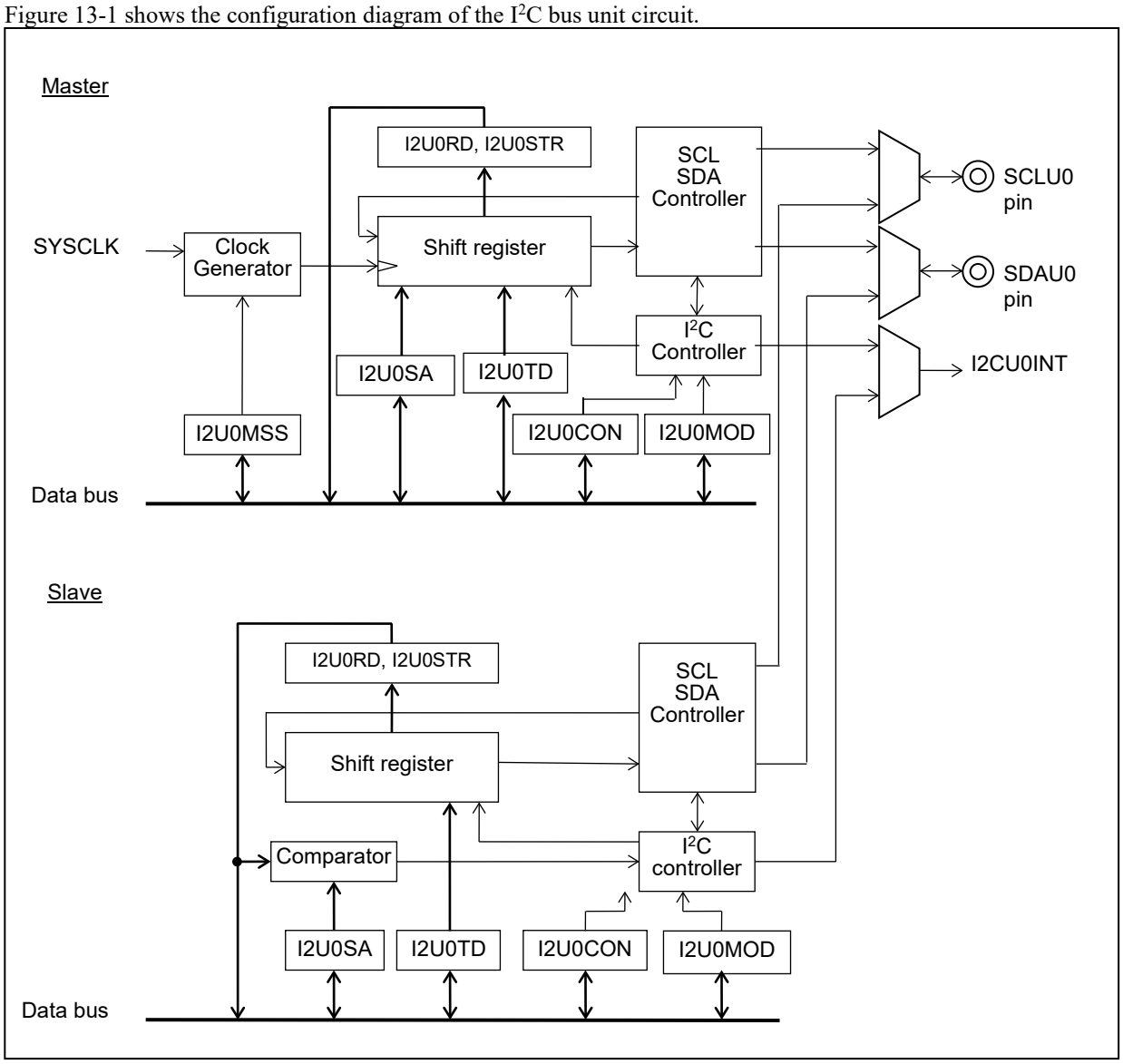
Table 13-1 shows features of I²C bus unit and I²C bus master.

Table 13-1 Features of I²C bus

Function	Operation mode	Features
I²C bus unit	Master function	- Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) - Support clock stretch function for the Slave - 7 or 10-bit address format - Self-test function by reading transmitted data onto the I²C bus (Safety function)
	Slave function	- Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) - Clock stretch function - 7-bit address format - Wake-up from STOP mode by matching slave address

13.1.2 Configuration

Figure 13-1 shows the configuration diagram of the I²C bus unit circuit.



SCLU0 : Serial clock
SDAU0 : Serial data
I2U0MSS : I²C bus unit 0 mode register
I2U0RD : I²C bus unit 0 receive register (Master/Slave Sharing)
I2U0SA : I²C bus unit 0 slave address register (Master/Slave Sharing)
I2U0TD : I²C bus unit 0 transmit data register (Master/Slave Sharing)
I2U0CON : I²C bus unit 0 control register (Master/Slave Sharing)
I2U0MOD : I²C bus unit 0 mode register (Master/Slave Sharing)
I2U0STR : I²C bus unit 0 status register (Master/Slave Sharing)
I2U0SCLR : I²C bus unit 0 status clear register (Master/Slave Sharing)

Figure 13-1 Configuration of I²C Bus Unit

13.1.3 List of Pins

The I/O pins of the I²C bus unit are assigned to the shared function of the general ports.

Pin name	I/O	Description
SDAU0	I/O	I²C bus unit 0 data I/O pin
SCLU0	I/O	I²C bus unit 0 clock I/O pin

Table 13-2 shows port used in the I²C and the register settings.
In addition to the mode setting of the shared function, select "Enable Input, Enable Output, N-ch open drain output and without pull-up" by setting following data to the port n mode register m (PnMODm).

Table 13-2 Port used in the I²C and the register settings

Channel No.	Pin name	Shared port		Setting register	Setting value	Available
0	SDAU0	P12	3rd Function	P1MOD2	0010_1011	●
	SCLU0	P13	3rd Function	P1MOD3	0010_1011	●

●: Available - : Unavailable

[Note]

- **Use external pull-up resistors for SDA pin and SCL pin according to the I²C bus specification. The internal pull-up resistors are unsatisfied the I²C bus specification. See the data sheet for each product for the value of internal pull-up resistors.**
- **If powering off this LSI in the slave mode, it disables communications of other devices on the I²C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.**
- **Do not connect multiple master devices on the I²C bus when using the master function.**

13.2 Description of Registers

13.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF780	I²C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF781	Reserved	-	-	-	-	-
0xF782	I²C bus unit 0 receive register	I2U0RD	-	R	8	0x00
0xF783	Reserved	-	-	-	-	-
0xF784	I²C bus unit 0 slave address register	I2U0SA	-	R/W	8	0x00
0xF785	Reserved	-	-	-	-	-
0xF786	I²C bus unit 0 transmit data register	I2U0TD	-	R/W	8	0x00
0xF787	Reserved	-	-	-	-	-
0xF788	I²C bus unit 0 control register	I2U0CON	-	R/W	8	0x00
0xF789	Reserved	-	-	-	-	-
0xF78A	I²C bus unit 0 mode register	I2U0MODL	I2U0MOD	R/W	8/16	0x00
0xF78B		I2U0MODH		R/W	8	0x22
0xF78C	I²C bus unit 0 status register	I2U0STAT	I2U0STR	R	8/16	0x00
0xF78D		I2U0ISR		R	8	0x00
0xF78E	I²C bus unit 0 status clear register	I2U0SCLRL	I2U0SCLR	W	8/16	0x00
0xF78F		I2U0SCLRH		W	8	0x00
0xF790	Reserved	-	-	-	-	-
0xF791	Reserved	-	-	-	-	-

13.2.2 I²C Bus Unit 0 Mode Register (I2U0MSS)

I2U0MSS is a SFR to select the Master mode or Slave mode of the I²C bus unit 0.

Address : 0xF780 (I2U0MSS)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0MSS							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2U0MD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	I2U0MD	Select the Master mode or Slave mode of the I²C but unit. 0: Master mode (Initial value) 1: Slave mode

[Note]

- **All SFRs are shared in master mode with slave mode. If switching master/slave mode, set "0" to I2U0EN bit of I2UMOD register, then switch the mode and reconfigurate each SFRs.**
- **Do not connect multiple master devices on the I²C bus when using the master function.**
- **If powering off this LSI in the slave mode, it disables communications of other devices on the I²C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.**

13.3 Description of Registers for Master function

This section explains about master mode of I²C bus unit 0.
In this section, both word symbol, byte symbol and bit symbol are put down with.
All SFRs are shared in master mode and slave mode. Reset master mode on I2U0MSS, at the begining.
When reset, the I²C bus unit is in master mode.
The initial values shown below are the values when set to master mode.

13.3.1 I²C Bus Unit 0 Receive Register (I2U0RD)

I2U0RD is a read-only SFR to store the received data.
This is initialized, in addition to reset function, by writing "0" to I2U0EN bit in I2U0MOD register.

Address : 0xF782(I2U0RD)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0RD							
Bit	-	-	-	-	-	-	-	-	I2U0R7	I2U0R6	I2U0R5	I2U0R4	I2U0R3	I2U0R2	I2U0R1	I2U0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is updated after completion of each reception.

Bit No.	Bit symbol name	Description
7 to 0	I2U0R7 to I2U0R0	Store the received data. This data is updated at coinciding slave-address and data reception. Reading this register enables following confirmation. - Reading when receiving data: Can confirm the received data. - Reading slave address or Reading when transmitting data: Can confirm the transmission data is surely transmitted.

13.3.2 I²C Bus Unit 0 Slave Address Register (I2U0SA)

I2U0SA is a SFR to set the address and transmission/reception mode of the slave device.
This is initialized, in addition to reset function, by writing "0" to I2U0EN bit in I2U0MOD register.

Address : 0xF784(I2U0SA)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0SA							
Bit	-	-	-	-	-	-	-	-	I2U0A6	I2U0A5	I2U0A4	I2U0A3	I2U0A2	I2U0A1	I2U0A0	I2U0RW
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2U0A6 to I2U0A0	Set the address of the communication partner.
0	I2U0RW	Select directions of the data communication. 0: Data transmission mode (Initial value) 1: Data reception mode

13.3.3 I²C Bus Unit 0 Transmit Data Register (I2U0TD)

2U0TD is a SFR to set the transmission data.
This is initialized, in addition to reset function, by writing "0" to I2U0EN bit in I2U0MOD register.

Address : 0xF786(I2U0TD)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0TD							
Bit	-	-	-	-	-	-	-	-	I2U0T7	I2U0T6	I2U0T5	I2U0T4	I2U0T3	I2U0T2	I2U0T1	I2U0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2U0T7 to I2U0T0	Set the transmission data.

13.3.4 I²C Bus Unit 0 Control Register (I2U0CON)

I2U0CON is a SFR to control transmission and reception operations.
This is initialized, in addition to reset function, by writing “0” to I2U0EN bit in I2U0MOD register.

Address : 0xF788(I2U0CON)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0CON							
Bit	-	-	-	-	-	-	-	-	I2U0ACT	-	-	-	-	I2U0RS	I2U0SP	I2U0ST
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2U0ACT	Set the acknowledgment data to be output at completion of reception. 0: Acknowledgment data "0" (Initial value) 1: Acknowledgment data "1"
6 to 3	-	Reserved bits
2	I2U0RS	This bit is a write-only and request a restart. When "1" is written to this bit during data communication, this module shifts to the restart condition and the communication restarts from the slave address. "1" can be written to this bit only while communication is active (I2U0ST = "1"). This bit always returns "0" for reading. 0: No restart request (Initial value) 1: Restart request
1	I2U0SP	This bit is a write-only and request a stop condition. When "1" is written to this bit, the module shifts to the stop condition and the communication stops. This bit always returns "0" for reading. 0: No stop condition request (Initial value) 1: Stop condition request
0	I2U0ST	Control the communication operation of this module. When "1" is written to this bit during I2U0ST bit is “0”, the communication starts. When "1" is overwritten to this bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, the data transmission/reception restarts. When "0" is written to this bit, the communication is stopped forcibly. When "1" is written to this bit, the I2U0ST bit is reset to “0”. 0: Stops communication (Initial value) 1: Starts communication

[Note]
- **Update it without a bit access instruction in the control register setting wait state. See Section 13.5.1.5 “Control Register Setting Wait State” for details.**
- **When I2U0ST bit is "1", write other bits of I2U0CON register in the control register setting wait state.**

13.3.5 I²C Bus Unit 0 Mode Register (I2U0MOD)

I2U0MOD is a SFR to set the operation mode.
See 13.5.4 “Operation Waveforms” for detail of communication speed.

Address : 0xF78A(I2U0MODL/I2U0MOD), 0xF78B(I2U0MODH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x2200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0MOD															
Byte	I2U0MODH								I2U0MODL							
Bit	I2U0TI2	I2U0TI1	I2U0TI0	-	-	-	I2U0CD1	I2U0CD0	-	-	I2U0MD4	I2U0MD3	I2U0MD2	I2U0MD1	I2U0MD0	I2U0EN
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	I2U0TI2 to I2U0TI0	Select the timing parameter for an I²C communication rate. See Table 13-3 of “13.5.4 Operation Waveforms” for detail. 000: Do not use (for SYSCLK = 16MHz) 001: a parameter for SYSCLK = 16MHz (Initial value) 010: Do not use (for SYSCLK = 16MHz) 011: a parameter for SYSCLK = 8MHz 100: a parameter for SYSCLK = 1MHz 101: a parameter for SYSCLK = LSCLK0 110: Do not use (for SYSCLK = 16MHz) 111: Do not use (for SYSCLK = 16MHz)
12 to 10	-	Reserved bits
9 to 8	I2U0CD1 to I2U0CD0	Select the I²C operating clock. 00: SYSCLK 01: 1/2 SYSCLK 10: 1/4 SYSCLK (Initial value) 11: 1/8 SYSCLK See Section 13.5.4 “Operation Waveforms” about communication speed.
7, 6	-	Reserved bits
5	I2U0MD4	Select whether to or not to use the clock stretch (handshake) function. This function monitors the I²C bus, therefore the communication speed gets lower depending on the load of I²C bus. 0: Not use the clock stretch function (Initial value) 1: Use the clock stretch function
4, 3	I2U0MD3 to I2U0MD2	Set the communication speed reduction rate. Specify these bits not so that the communication speed exceeds 100 kbps/400kbps/1 Mbps. When parameter for LSCLK0 or 1MHz is selected by the I2U0TI2-0 bits, "No communication speed reduction" is selected regardless this setting. See Table 13-3 of “13.5.4 Operation Waveforms” for detail. 00: No communication speed reduction (Initial value) 01: Approx. 10% communication speed reduction 10: Approx. 17% communication speed reduction 11: Approx. 24% communication speed reduction
2, 1	I2U0MD1 to I2U0MD0	Set the communication speed mode. 00: Standard mode (Initial value) (100 kbps*) 01: Fast mode (400 kbps*) 10: 1Mbps mode (1Mbps*) 11: 1Mbps mode (1Mbps*) * : When SYSCLK=16 MHz and I2U0CD1-0 = “00” and I2U0MD4 = “0” and I2U0TI2-0 = “000/001” and I2U0MD3-2 = “00”.

Bit No.	Bit symbol name	Description
0	I2U0EN	Enable the master operation. When "1" is written to this bit, the I2U0ST bit can be set. When "0" is written to this bit, the I²C master stops operation and the I2U0RD, I2U0SA, I2U0TD, I2U0CON and I2U0STR are initialized. If this bit is written “0” during a communication, do initialization and re-setting. 0: Stop the I²C master operation (Initial value) 1: Enable the I²C master operation

13.3.6 I²C Bus Unit 0 Status Register (I2U0STR)

I2U0STR is a SFR to indicate the state of the I²C bus unit.
This is initialized, in addition to reset function, by writing "0" to I2U0EN bit in I2U0MOD register.
Each bit is initialized by writing "1" to a corresponding bit of I2U0SCLR register.

Address : 0xF78C(I2U0STAT/I2U0STR), 0xF78D(I2U0ISR)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0STR															
Byte	I2U0ISR								I2U0STAT							
Bit	-	-	-	-	-	I2U0SPS	I2U0DS	I2U0AS	rsvd	-	-	-	-	I2U0ER	I2U0ACR	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	I2U0SPS	Indicate the usage state of the I²C bus. This bit is set to "1" when transmitting the stop condition has been completed on the I²C bus. To reset this bit, write "1" to I2U0CSPS bit of I2U0SCLR register. 0: The stop condition has not been transmitted (Initial value) 1: The stop condition has been transmitted
9	I2U0DS	Indicate the usage state of the I²C bus. This bit is set to "1" when transmitting data or receiving data has been completed on the I²C bus. To reset this bit, write "1" toI2U0CDS bit of I2U0SCLR register. 0: The transmission/reception has not been completed (Initial value) 1: The transmission/reception has been completed
8	I2U0AS	Indicate the usage state of the I²C bus. This bit is set to "1" when transmitting the start condition and 7 bit slave address have been completed on the I²C bus. To reset this bit, write "1" to I2U0CAS bit of I2U0SCLR register. 0: The start condition and the slave address have not been transmitted (Initial value) 1: The start condition and the slave address have been transmitted
7	rsvd	Reserved bit
6 to 3	-	Reserved bits
2	I2U0ER	Indicate a transmission error. When a bit of transmission data and the value on the SDAU0 pin do not coincide, "1" is set to this bit. To reset this bit, write "1" toI2U0CER bit of I2U0SCLR register. When this bit is set to "1" and the clock stretch function is used (I2U0MD4 = "1"), the SDAU0 pin output is disabled until the subsequent byte data communication terminates. Even if this bit is set to "1", the SDAU0 pin output continues until the subsequent byte data communication terminates when the clock stretch function is not used (I2U0MD4 = "0"). 0: There was no transmission error (Initial value) 1: There was a transmission error
1	I2U0ACR	Store the acknowledgment signal received. Acknowledgment signals are received when the slave address is transmitted and the data transmission/reception is completed. To reset this bit, write "1" to I2U0CACR bit of I2U0SCLR register. 0: Received acknowledgment "0" (Initial value) 1: Received acknowledgment "1"
0	-	Reserved bit

13.3.7 I²C Bus Unit 0 Status Clear Register (I2U0SCLR)

I2U0SCLR is a SFR to clear the state of the I²C bus unit.
When Each bit is written "1", a corresponding bit of I2U0STR register is initialized to "0".

Address : 0xF78E(I2U0SCLRL/I2U0SCLR), 0xF78F(I2U0SCLRH)
Access : W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0SCLR															
Byte	I2U0SCLRH								I2U0SCLRL							
Bit	-	-	-	-	-	I2U0CSPS	I2U0CDS	I2U0CAS	-	-	-	-	-	I2U0CER	I2U0CACR	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to clear a target status.
Writing "0": Invalid
Writing "1": Clear a target interrupt

Bit No.	Bit symbol name	Description (target)
15 to 11	-	Reserved bits
10	I2U0CSPS	I2U0SPS bit of I2U0STR register
9	I2U0CDS	I2U0DS bit of I2U0STR register
8	I2U0CAS	I2U0AS bit of I2U0STR register
7 to 3	-	Reserved bits
2	I2U0CER	I2U0ER bit of I2U0STR register
1	I2U0CACR	I2U0ACR bit of I2U0STR register
0	-	Reserved bit

13.4 Description of Registers for Slave function

This section explains about slave mode of I²C bus unit 0.
In this section, both word symbol, byte symbol and bit symbol are put down with.
All SFRs are shared in master mode and slave mode. Set slave mode on I2U0MSS, at the beginning.
When reset, the I²C bus unit is in master mode.
The initial values shown below are the values when set to slave mode.

13.4.1 I²C Bus Unit 0 Receive Register (I2U0RD)

I2U0RD is a read-only SFR used to store the received data.

Address : 0xF782(I2U0RD)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0RD							
Bit	-	-	-	-	-	-	-	-	I2U0R7	I2U0R6	I2U0R5	I2U0R4	I2U0R3	I2U0R2	I2U0R1	I2U0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2U0R7 to I2U0R0	Store the received data. This data is updated at coinciding slave-address and data reception. Reading this register enables following confirmation. - Reading when receiving data: Can confirm the received data. - Reading when transmitting data: Can confirm the transmission data is surely transmitted.

13.4.2 I²C Bus Unit 0 Slave Address Register (I2U0SA)

I2U0SA is a SFR to set the slave address.

Address : 0xF784(I2U0SA)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0SA							
Bit	-	-	-	-	-	-	-	-	I2U0A6	I2U0A5	I2U0A4	I2U0A3	I2U0A2	I2U0A1	I2U0A0	-
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2U0A6 to I2U0A0	Set the slave address.
0	-	Reserved bit

13.4.3 I²C Bus Unit 0 Transmit Data Register (I2U0TD)

I2U0TD is a SFR to set the transmission data.

Address : 0x786(I2U0TD)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0TD							
Bit	-	-	-	-	-	-	-	-	I2U0T7	I2U0T6	I2U0T5	I2U0T4	I2U0T3	I2U0T2	I2U0T1	I2U0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2U0T7 to I2U0T0	Set the transmission data.

13.4.4 I²C Bus Unit 0 Control Register (I2U0CON)

I2U0CON is a SFR to control transmission and reception operations.

Address : 0xF788(I2U0CON)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0CON							
Bit	-	-	-	-	-	-	-	-	I2U0ACT	-	I2U0WT	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R/W	R	W	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2U0ACT	Set the acknowledgment data to be output at completion of reception in the slave mode. 0: Acknowledgment data "0" (Initial value) 1: Acknowledgment data "1"
6	-	Reserved bit
5	I2U0WT	Release the communication wait state ("L" level output on the SCLU0 pin) in the slave mode. Writing "1" to this bit during the communication wait state releases the state ("L" level output of the SCLU0 pin is released). This bit is a write-only bit and always returns "0" for reading. 0: Not release the communication wait state (Initial value) 1: Release the communication wait state
4 to 0	-	Reserved bits

[Note]
- **If system clock is extremely slower than the communication speed, the data transmission/reception can be failed.**
- **Before releasing the communication wait state, change the system clock enough speed for the communication.**

13.4.5 I²C Bus Unit 0 Mode Register (I2U0MOD)

I2U0MOD is a SFR used to set the operation mode.

Address : 0xF78A(I2U0MODL/I2U0MOD), 0xF78B(I2U0MODH),
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000 *1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0MOD															
Byte	I2U0MODH								I2U0MODL							
Bit	-	-	-	-	-	-	-	-	-	I2U0MD5	I2U0MD4	I2U0MD3	I2U0MD2	I2U0MD1	-	I2U0EN
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*1: This is the initial value when the I²C bus unit is set to slave mode (I2U0MSS=1).

Bit No.	Bit symbol name	Description
15 to 7	-	Reserved bits
6	I2U0MD5	Enable or disable the start condition interrupt in the slave mode. 0: Disabled (Initial value) 1: Enabled
5	I2U0MD4	Enable or disable the stop condition interrupt in the slave mode. 0: Disabled (Initial value) 1: Enabled
4	I2U0MD3	Control enable or disable the slave address unmatched interrupt while communicating to the master, receiving re-start condition and another slave is selected. This function performs detecting the status of I2US0SAA bit. Do not clear the I2US0SAA bit by the software when enabling the interrupt. 0: Disabled (Initial value) 1: Enabled
3	I2U0MD2	Enable or disable the communication wait function of the I²C bus unit (output "L" level on the SCLU0 pin) when transmitting to the master and receiving the acknowledge data "1" from the master. Set this bit to "1" to use the communication wait function. 0: Disabled (Initial value) 1: Enabled
2	I2U0MD1	Select mode when the stop condition interrupt is enabled (I2U0MD4=1). This function performs detecting the status of I2US0SAA bit. Do not clear I2US0SAA bit by the software when enabling the interrupt. 0: The interrupt occur while the master is communicating with self-slave or other slaves (Initial value) 1: The interrupt occur while the master is communicating with only self-slave
1	-	Reserved bit
0	I2U0EN	Enable the slave operation of the I²C bus unit. When "1" is written to this bit, the operation of the I²C bus unit 0 is enabled. When "0" is written to this bit, all the bits of the I²C bus status register (I2US0STR) are initialized to "0", and the operation of the I²C bus unit 0 is stopped. 0: Stop the I²C slave operation (Initial value) 1: Enable the I²C slave operation

[Note]

- **To be disable the wake-up from standby mode by matching the slave address, Stop the operation by resetting I2U0EN bit to "0" before entering STOP mode.**

13.4.6 I²C Bus Unit 0 Status Register (I2U0STR)

I2U0STR is a SFR to indicate the state of the I²C bus unit.
Each bit is initialized, in addition to reset function, by writing "1" to a corresponding bit of I2U0SCLR register.

Address : 0xF78C(I2U0STAT/I2U0STR), 0xF78D(I2U0ISR)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0STR															
Byte	I2U0ISR								I2U0STAT							
Bit	-	-	I2U0ASNA	I2U0RAS	I2U0STS	I2U0SPS	I2U0DS	I2U0AS	-	-	-	I2U0TR	I2U0SAA	I2U0ER	I2U0ACR	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 14	-	Reserved bits
13	I2U0ASNA	Reserved bit. Fixed to "0"
12	I2U0RAS	Indicate status of the interrupt when enabling the start condition interrupt (I2U0MD3 bit = 1). To reset this bit, write "1" to I2U0CRAS bit of I2U0SCLR register. 0: Unmatched the slave address is detected after the start condition (Initial value) 1: Unmatched the slave address is detected after the start condition
11	I2U0STS	Indicate status of transmission and reception. This bit is set to "1" when receiving the start condition. This bit is available when I2U0MD5 bit is "1". To reset this bit, write "1" to I2U0CSTS bit of I2U0SCLR register. 0: The start condition has not been received (Initial value) 1: The start condition has been received
10	I2U0SPS	Indicate status of transmission and receive. This bit is set to "1" when receiving the stop condition. This bit is available when I2U0MD4 bit is "1". To reset this bit, write "1" to I2U0CSPS bit of I2U0SCLR register. 0: The stop condition has not been received (Initial value) 1: The stop condition has been received
9	I2U0DS	Indicate status of transmission and reception. This bit is set to "1" when transmitting or receiving data on the condition of that slave address is matched. To reset this bit, write "1" to I2U0CDS bit of I2U0SCLR register. 0: The data has not been transmitted or received (Initial value) 1: The data has been transmitted or received
8	I2U0AS	Indicate status of transmission and reception. This bit is set to "1" when receiving the slave address data and it is matched. To reset this bit, write "1" to I2U0CAS bit of I2U0SCLR register. 0: The slave address has not been received or it is not matched (Initial value) 1: The slave address has been received and it is matched
7 to 5	-	Reserved bits
4	I2U0TR	Indicate the transmitting or receiving state. This bit is set to "1" when detecting the data reception mode. This bit is reset to "0" when detecting a stop condition or detecting the data transmission mode. To reset this bit, write "1" to I2U0CTR bit of I2U0SCLR register. 0: Receiving state (Initial value) 1: Transmitting state
3	I2U0SAA	Indicate that this device is specified as a slave address. This bit is set to "1" when the content of the slave address output by the master device coincides with the contents of I2US0SA register This bit is reset to "0" when a stop condition is received or when "1" is written to I2U0CSAA bit of I2U0SCLR register. 0: Not coincide with the slave address (Initial value) 1: Coincides with the slave address

Bit No.	Bit symbol name	Description
2	I2U0ER	Indicate a transmission error. When the value of the bit transmitted and the value of the SDAU0 pin do not coincide, this bit is set to "1". When this bit is set to "1", the SDAU0 pin output is disabled until the subsequent byte data communication terminates. To reset this bit, write "1" toI2U0CER bit of I2U0SCLR register. 0: There was no transmission error (Initial value) 1: There was a transmission error
1	I2U0ACR	Store an acknowledgment signal received. The acknowledgment signals are received each time the slave address is received and data transmission or reception is completed. To reset this bit, write "1" to I2U0CTR bit of I2U0SCLR register. 0: Received the acknowledgment "0" (Initial value) 1: Received the acknowledgment "1"
0	-	Reserved bit

13.4.7 I²C Bus Unit 0 Status Clear Register (I2U0SCLR)

I2U0SCLR is a SFR to clear the state of the I²C bus unit.
When Each bit is written "1", a corresponding bit of I2U0STR/I2M0STR register is initialized to "0".

Address : 0xF78E(I2U0SCLRL/I2U0SCLR), 0xF78F(I2U0SCLRH)
Access : W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0SCLR															
Byte	I2U0SCLRH								I2U0SCLRL							
Bit	-	-	I2U0CASNA	I2U0CRAS	I2U0CSTS	I2U0CSPS	I2U0CDS	I2U0CAS	-	-	-	I2U0CTR	I2U0CSAA	I2U0CER	I2U0CACR	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to clear a target status.
Writing "0": Invalid
Writing "1": Clear a target interrupt

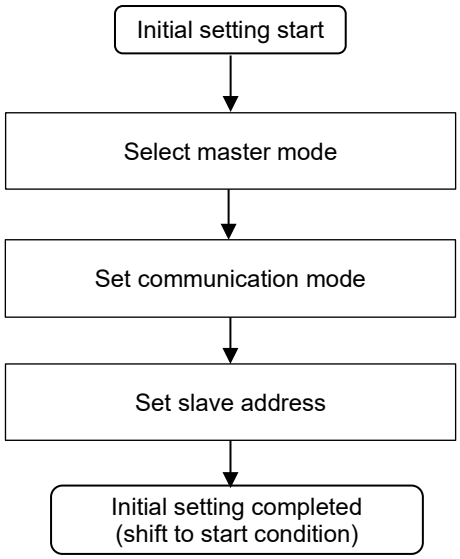
Bit No.	Bit symbol name	Description (Target status)
15 to 14	-	Reserved bits
13	I2U0CASNA	I2U0ASNA bit of I2U0STR register
12	I2U0CRAS	I2U0RAS bit of I2U0STR register
11	I2U0CSTS	I2U0STS bit of I2U0STR register
10	I2U0CSPS	I2U0SPS bit of I2U0STR register
9	I2U0CDS	I2U0DS bit of I2U0STR register
8	I2U0CAS	I2U0AS bit of I2U0STR register
7 to 5	-	Reserved bits
4	I2U0CTR	I2U0TR bit of I2U0STR register
3	I2U0CSAA	I2U0SAA bit of I2U0STR register
2	I2U0CER	I2U0ER bit of I2U0STR register
1	I2U0CACR	I2U0ACR bit of I2U0STR register
0	-	Reserved bit

13.5 Description of Operation for Master function

13.5.1 Control Procedures

The following flow charts describe procedures of each operation in the master mode.

13.5.1.1 Initial Setting of Communication Operation

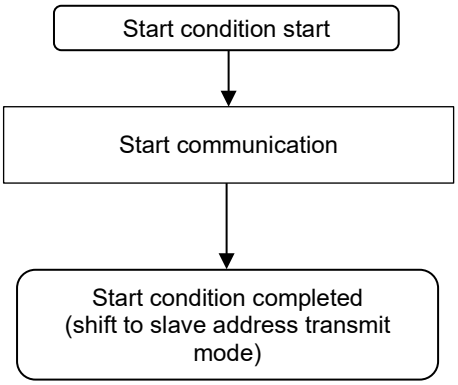


Set I2U0MD="0" in the I2U0MSS register

Set I2U0EN="1" in the I2U0MOD register

Set I2U0A6 to I2U0A0 bits and I2U0RW bit in the I2U0SA register

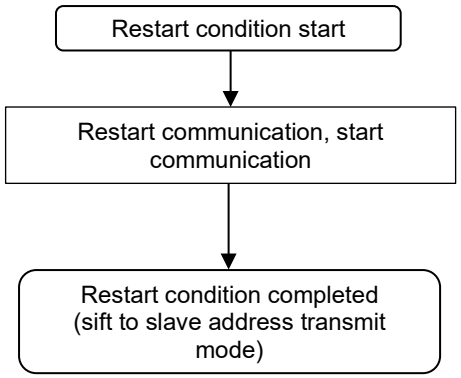
13.5.1.2 Start Condition



Set I2U0ST bit of I2U0CON register to "1".

Output the start condition waveforms to SDAU0 and SCLU0 pins.

13.5.1.3 Restart Condition

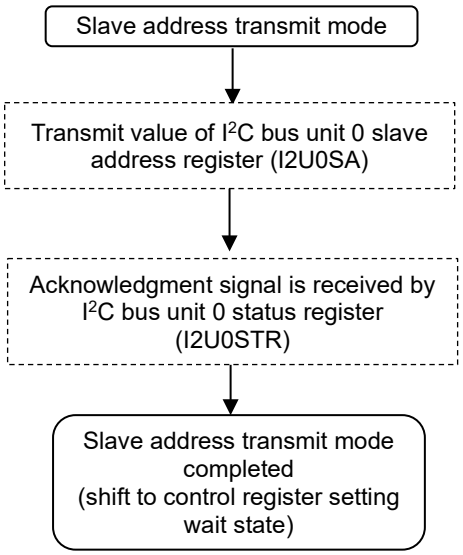


Communication in progress (I2U0ST="1")

Set I2U0RS="1" and I2U0ST="1" in the I2U0CON register

Output restart condition waveforms to SDAU0 and SCLU0 pins.

13.5.1.4 Slave Address Transmission Mode

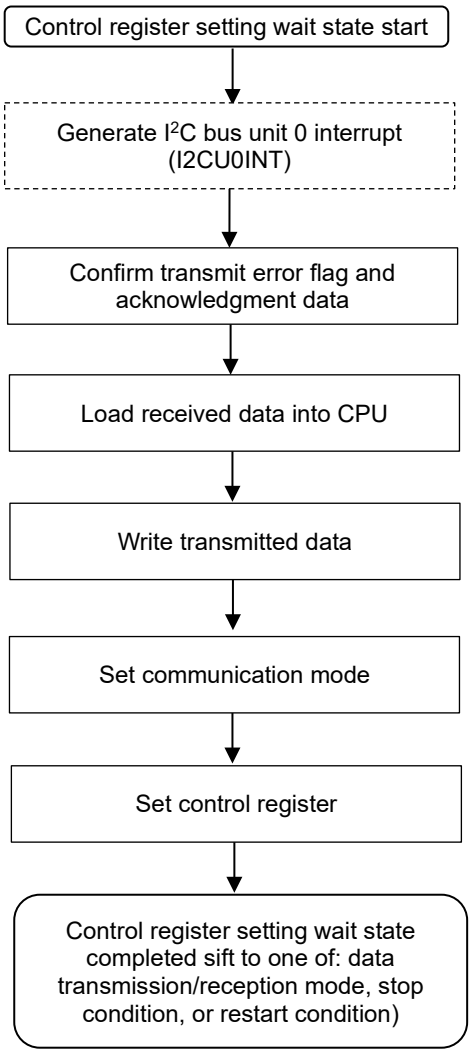


The value is transmitted from SDAU0 pin in MSB first through hardware following the start condition
I2U0A6 to I2U0A0 bits: Slave address
I2U0RW: Data direction (transmission/reception)
Value transmitted from the SDAU0 pin is stored in the I2U0RD register

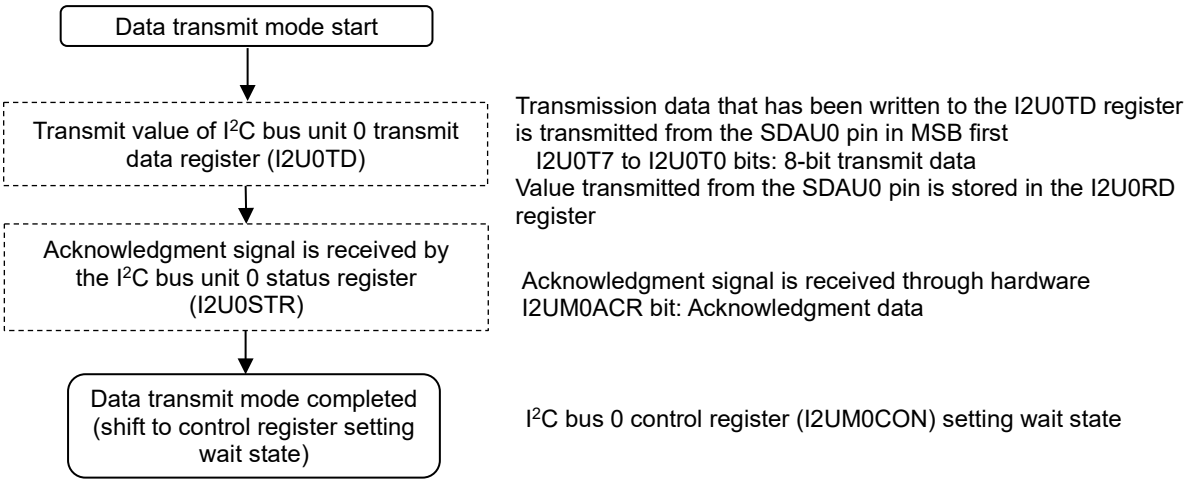
Acknowledgment signal is received through hardware
I2UM0ACR bit: Acknowledgment data

I²C bus 0 control register (I2UM0CON) setting wait state

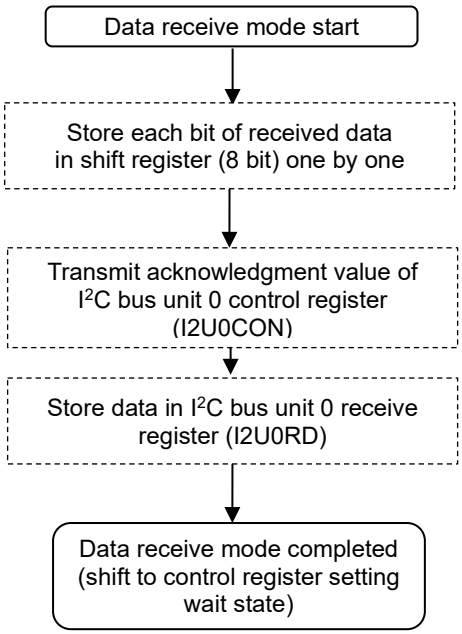
13.5.1.5 Control Register Setting Wait State



13.5.1.6 Data Transmission Mode



13.5.1.7 Data Reception Mode



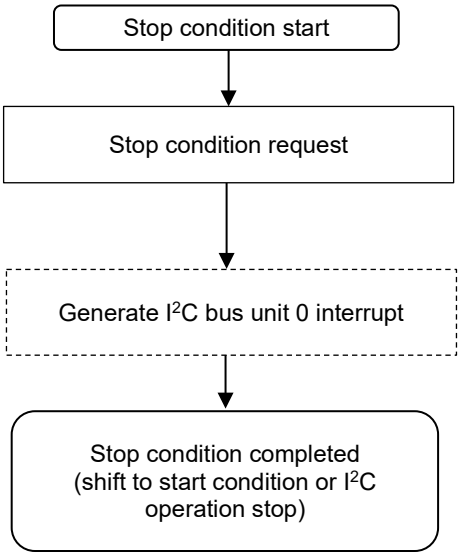
Value (received data) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Acknowledgment signal is transmitted through hardware
I2U0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2U0ACR bit of the I2U0STA register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2U0R7 to I2U0R0 bits: 8-bit receive data

I²C bus unit 0 control register (I2U0CON) setting wait state

13.5.1.8 Stop Condition



Set I2U0SP bit of I2U0CON register to “1”.

Output stop condition waveforms to SDAU0 and SCLU0 pins.

After the stop condition waveform is output, an interrupt is generated through hardware

13.5.2 Communication Operation Timing

Figures 13-2 to 13-4 show the operation timing and control method for each communication mode during the master operation.

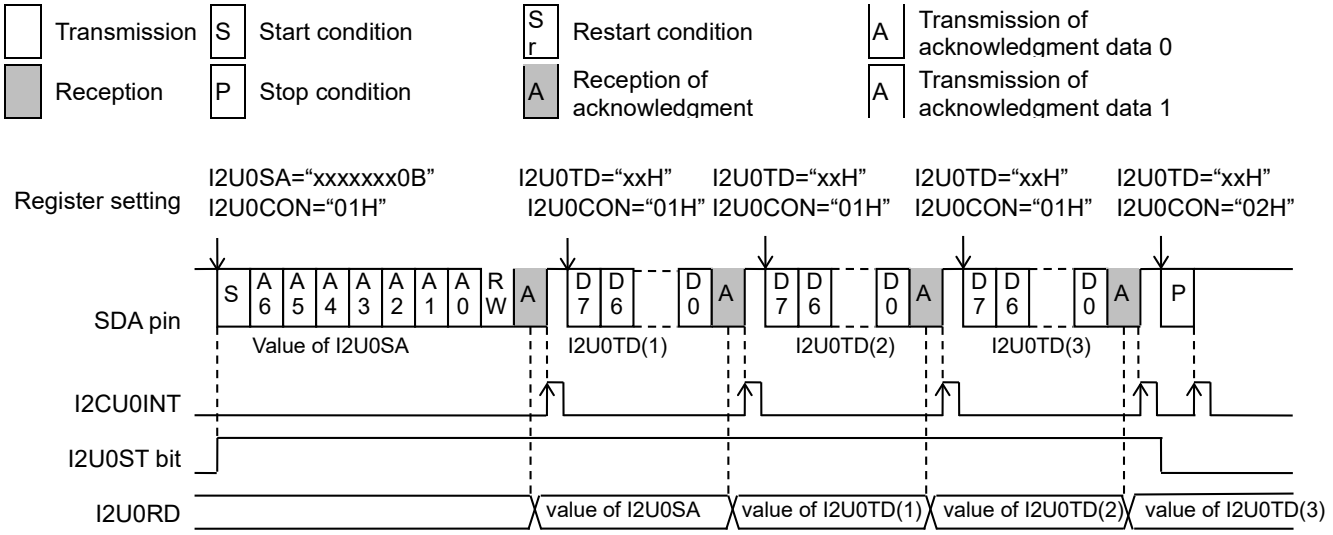


Figure 13-2 Operation timing during data transmission in the master mode

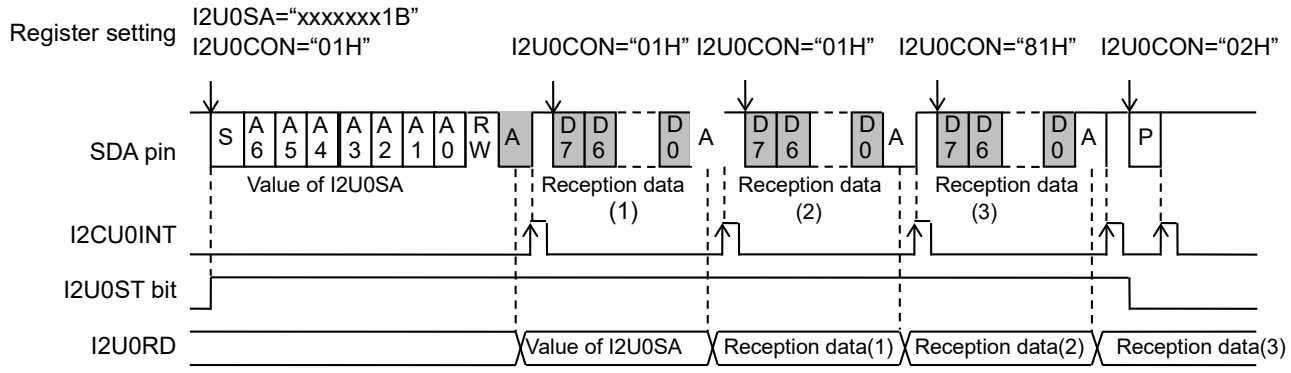


Figure 13-3 Operation timing during data reception in the master mode

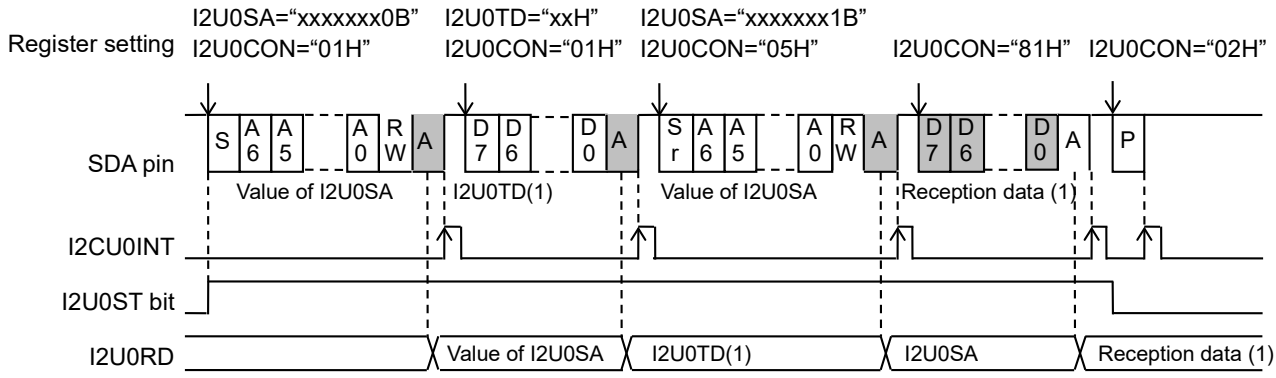


Figure 13-4 Operation timing during data transmission/ reception in the master mode

Figure 13-5 shows the operation timing and control method when an acknowledgment error occurs.

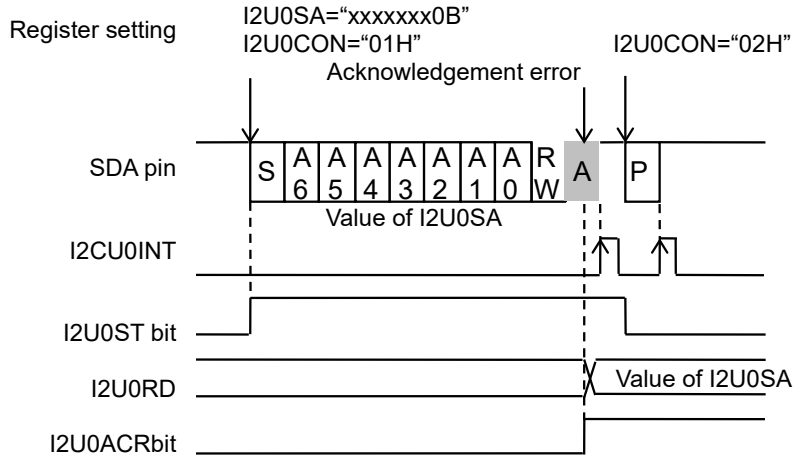


Figure 13-5 Operation suspend timing at occurrence of acknowledgment error in the master mode

When the values of the transmitted bit and the SDAU0 pin do not coincide, the I2U0ER bit of the I²C bus unit 0 status register (I2U0STR) is set to "1" and the SDAU0 pin output is disabled until termination of the subsequent byte data communication.

Figure 13-6 shows the operation timing and control method when transmission fails.

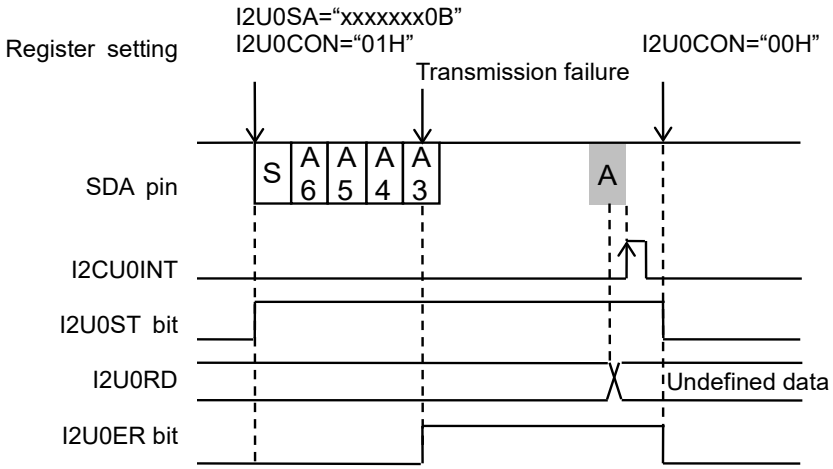


Figure 13-6 Operation timing when transmission fails in the master mode

13.5.3 Interrupt

The following is interrupt causes in master operation.

Interrupt causes	Timing that the interrupt is occurred
Transmission of a slave address	At entry to control register setting wait state after end of slave address transmission mode
Data transmission	At entry to control register setting wait state after end of data transmission mode
Data reception	At entry to control register setting wait state after end of data reception mode
Output stop condition	After output stop condition waveform and passing t_{BUF}.

13.5.4 Operation Waveforms

Figure 13-7 shows the operation waveforms of SDAU0 and SCLU0 pins. Table 13-3 shows the relationship between communication speeds and I²C operating clock counts.

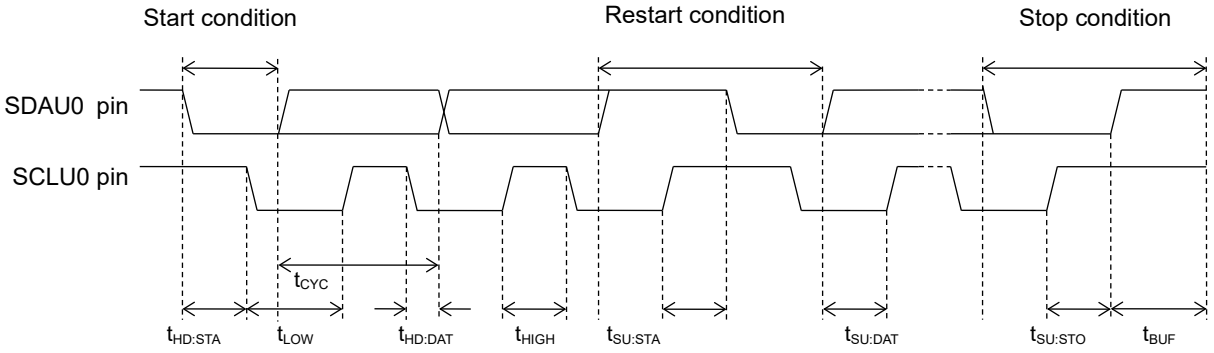


Figure 13-7 Operation Waveforms of SDAU0 and SCLU0 Pins

Table 13-3 relationship between communication speeds and I²C operating clock counts.

I2U0MOD register			AC timing [I²C operating clock counts]									I²C operating frequency and communication speed [kHz]			
I2U0TI2-0	I2U0MD1-0	I2U0MD3-2	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}	16MHz	8MHz	1MHz	LSCLK0
0	0	0	160	72	88	16	72	88	72	72	88	100.0	50.0	6.3	0.2
0	0	1	176	80	96	16	80	96	80	80	96	90.9	45.5	5.7	0.2
0	0	2	192	88	104	16	88	104	88	88	104	83.3	41.7	5.2	0.2
0	0	3	208	96	112	16	96	112	96	96	112	76.9	38.5	4.8	0.2
0	1	0	40	14	26	12	14	26	14	14	26	400.0	200.0	25.0	0.8
0	1	1	44	16	28	12	16	28	16	16	28	363.6	181.8	22.7	0.7
0	1	2	48	18	30	12	18	30	18	18	30	333.3	166.7	20.8	0.7
0	1	3	52	20	32	12	20	32	20	20	32	307.7	153.9	19.2	0.6
0	2	0	16	6	10	4	6	10	6	6	10	1000.0	500.0	62.5	2.0
0	2	1	18	7	11	4	7	11	7	7	11	888.9	444.5	55.6	1.8
0	2	2	19	8	11	4	8	11	7	8	11	842.1	421.1	52.6	1.7
0	2	3	21	9	12	4	9	12	8	9	12	761.9	381.0	47.6	1.6
1	0	0	80	36	44	8	36	44	36	36	44	200.0	100.0	12.5	0.4
1	0	1	88	40	48	8	40	48	40	40	48	181.8	90.9	11.4	0.4
1	0	2	96	44	52	8	44	52	44	44	52	166.7	83.4	10.4	0.3
1	0	3	104	48	56	8	48	56	48	48	56	153.8	76.9	9.6	0.3
1	1	0	20	7	13	6	7	13	7	7	13	800.0	400.0	50.0	1.6
1	1	1	22	8	14	6	8	14	8	8	14	727.3	363.7	45.5	1.5
1	1	2	24	9	15	6	9	15	9	9	15	666.7	333.4	41.7	1.4
1	1	3	26	10	16	6	10	16	10	10	16	615.4	307.7	38.5	1.3
1	2	0	8	3	5	2	3	5	3	3	5	2000.0*	1000.0	125.0	4.1
1	2	1	9	4	5	2	4	5	3	4	5	2000.0*	1000.0	125.0	4.1
1	2	2	10	4	6	2	4	6	4	4	6	1777.8*	888.9	111.1	3.6
1	2	3	11	5	6	2	5	6	4	5	6	1600.0*	800.0	100.0	3.3
2	0	**	10	5	5	1	5	5	4	5	5	1600.0*	800.0	100.0	3.3
2	1	**	4	2	2	1	2	2	1	2	2	4000.0*	2000.0*	250.0	8.2
2	2	**	4	2	2	1	2	2	1	2	2	4000.0*	2000.0*	250.0	8.2
3	0	**	16	8	8	1	8	8	7	8	8	1000.0*	500.0	62.5	2.0
3	1	**	8	4	4	1	4	4	3	4	4	2000.0*	1000.0	125.0	4.1
3	2	**	4	2	2	1	2	2	1	2	2	4000.0*	2000.0*	250.0	8.2

*: The operation is not guaranteed when over 1MHz speed.
**: The setting is invalid.

These clock counts is in case of I2U0CD1-0 bits = “00”. If it is not “00”, its counts increase depending on dividing rate.

[Note]

- **When the slave device uses the clock stretch function which holds SCLU0 pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.**

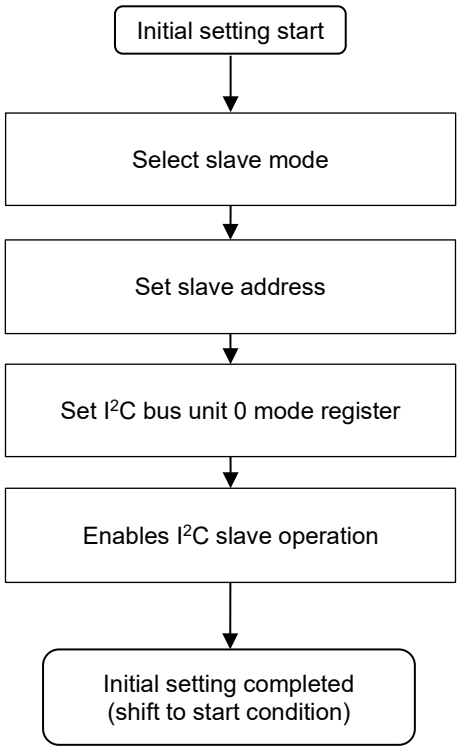
13.6 Description of Operation for Slave function

I²C bus unit 0 only have slave function.

13.6.1 Procedures

The following flow charts describe procedures of each operation in the slave mode.

13.6.1.1 Initial Setting of Communication Operation



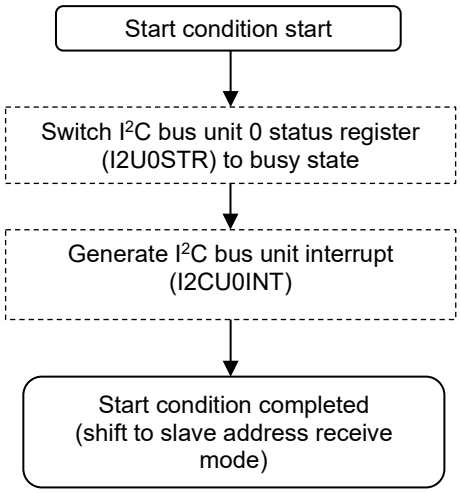
Select I2U0MD= “1” in the I2U0MSS register

Set I2U0A6 to I2U0A0 bits in the I2U0SA register

Set the I2U0MD register
I2U0MD5 bit: Enable/disable start condition interrupt
I2U0MD4 bit: Enable/disable stop condition interrupt

Set I2U0EN=“1” with the I2U0MD register

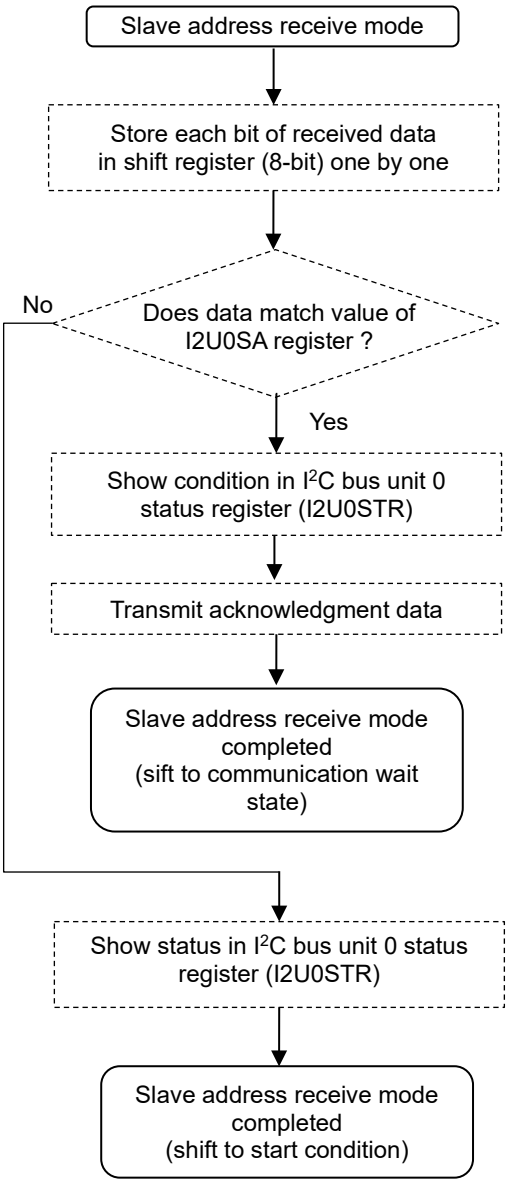
13.6.1.2 Start Condition



Value of the following bit becomes "1" when start condition waveforms are input to SDAU0 and SCLU0 pins

<Only when I2U0MD5=1>
Start condition interrupt is generated

13.6.1.3 Slave Address Reception Mode



Value (slave address, data direction bit) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Compare slave address stored in shift register with slave address of I2U0SA

I2U0SAA bit: Data matches slave address (I2U0SAA=1)
I2U0TR bit: Transmit state/receive state

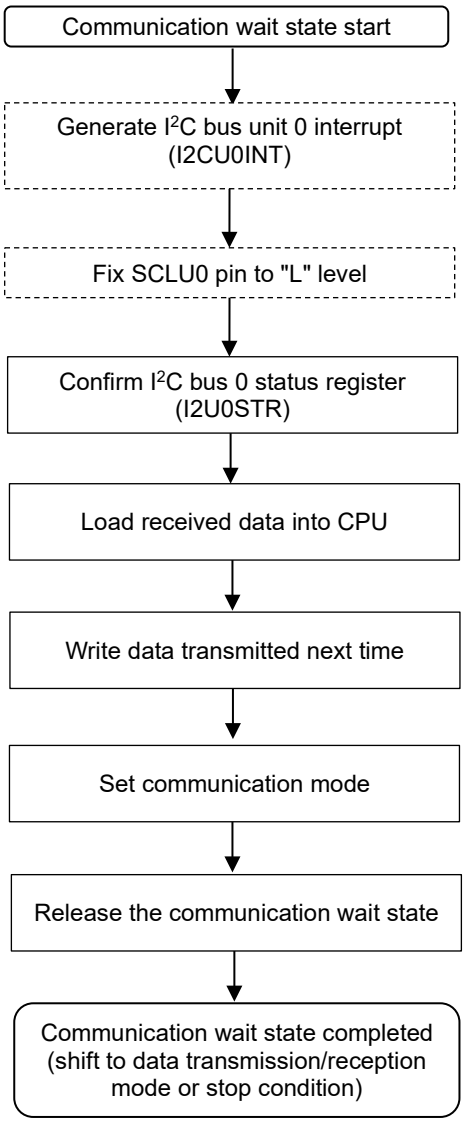
Acknowledgment data ("L" level) is output

Shift to the communication wait state after detection of falling edge of transfer clock which was input to SCLU0 pin transmitting acknowledgment data

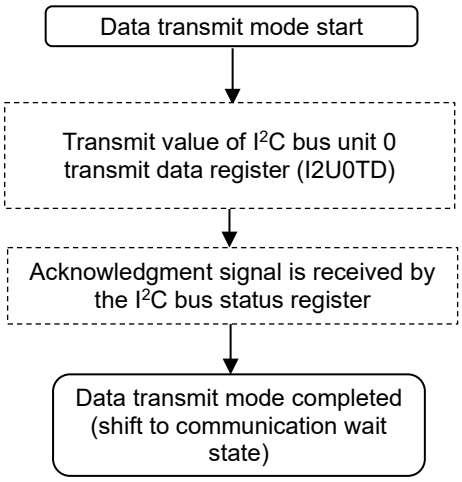
I2U0SAA bit: Data does not match slave address (I2U0SAA=0)

If not match, neither acknowledgment nor interrupt is generated

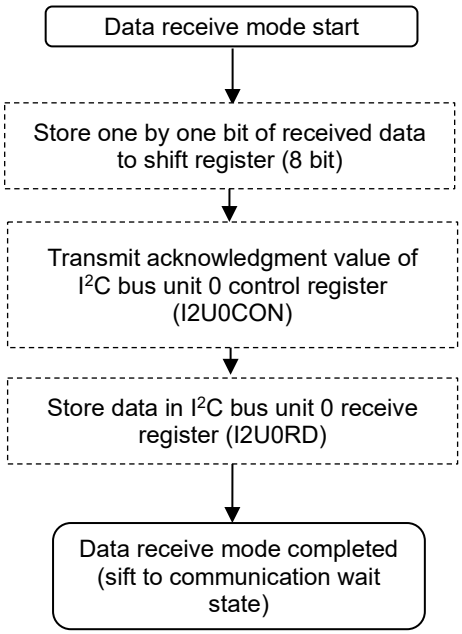
13.6.1.4 Communication Wait State



13.6.1.5 Data Transmission Mode



13.6.1.6 Data Reception Mode



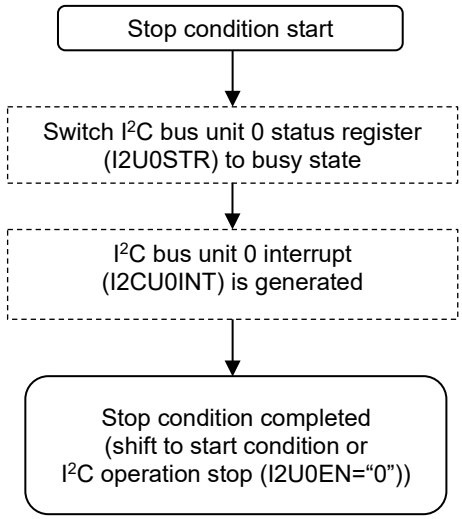
Value (received data) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Acknowledgment signal is transmitted through hardware
I2U0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2U0ACR bit of the I2U0STR register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2U0R7 to I2U0R0 bits: 8-bit receive data

Shift to communication wait state after detection of falling edge of transfer clock which was input to SCLU0 pin transmitting acknowledgment data

13.6.1.7 Stop Condition

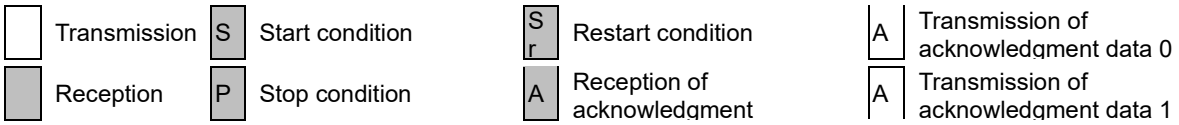


Value of the following bit becomes "0" when stop condition waveforms are input to SDAU0 and SCLU0 pins.

<Only when I2U0MD4="1">
Stop condition interrupt is generated

13.6.2 Communication Operation Timing

Figures 13-8 to 13-10 show the operation timing and control method for each communication mode.



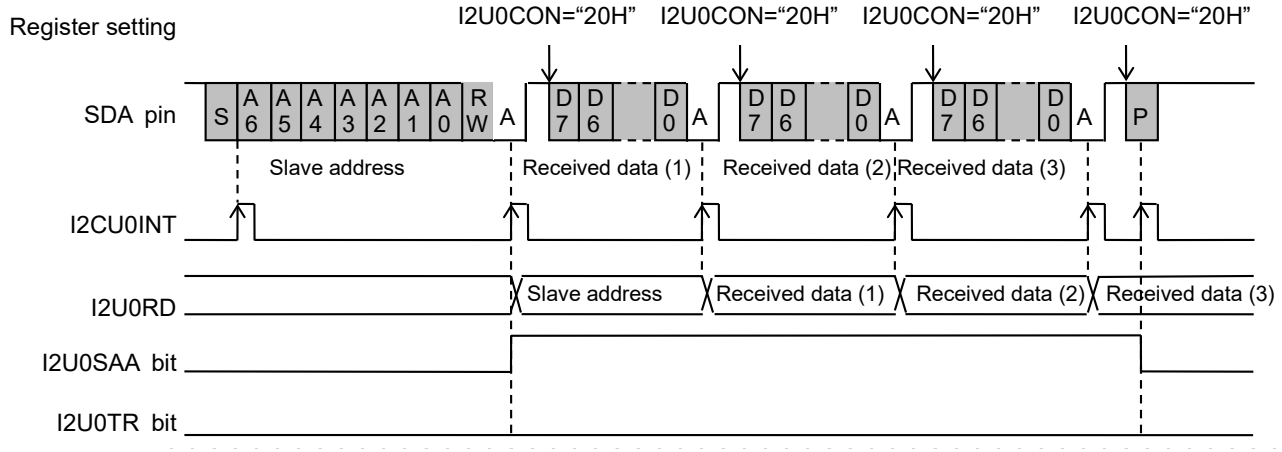


Figure 13-8 Operation Timing in Data Reception Mode When Slave Mode is selected

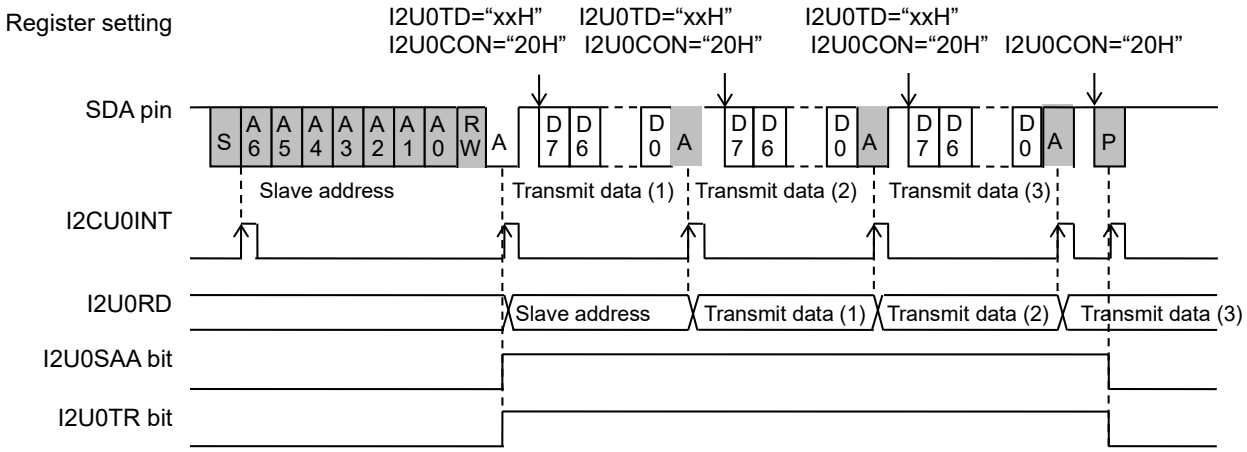


Figure 13-9 Operation Timing in Data Transmission Mode When Slave Mode is selected

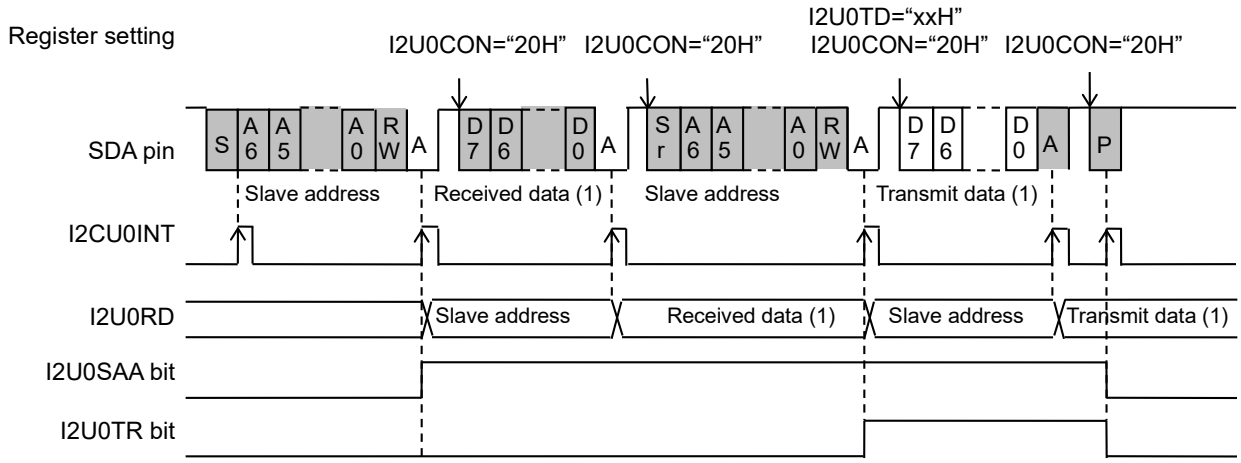


Figure 13-10 Operation Timing at Data Transmission/Reception Mode Switching When Slave Mode is selected

When the values of the transmitted bit and the SDAU0 pin do not coincide, the I2U0ER bit of the I²C bus unit 0 status register (I2U0STR) is set to "1" and the SDAU0 pin output is disabled until termination of the subsequent byte data communication.
Figure 13-11 shows the operation timing and control method when transmission fails.

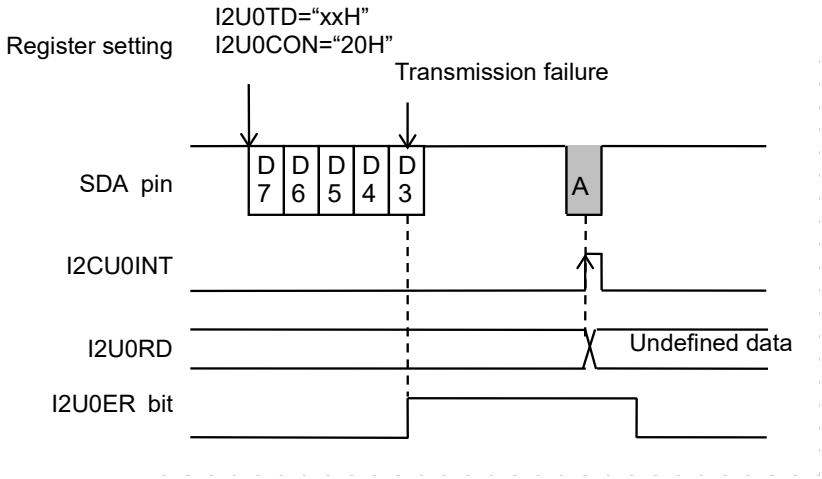


Figure 13-11 Operation Timing When Transmission Fails When Slave Mode is selected

[Note]
- **If entering to the STOP mode while the slave mode is enabled, first make sure that communication is not in progress (from coincidence of address to reception of stop condition).**

13.6.3 Interrupt

Table 13-4 shows interrupt causes in slave operation.

Table 13-4 List of slave interrupt

Interrupt causes	Setting to enable	Status flag in the I2U0STR register	Timing that the interrupt is occurred
Start condition	I2U0MD5= "1"	I2U0STS= "1"	After output start condition waveform.
Coinciding Slave Address	-	I2U0AS= "1"	At entry to control register setting wait state with coinciding slave-address after end of slave address reception mode
Data transmission	-	I2U0DS= "1"	At entry to control register setting wait state after end of data transmission mode
Data reception	-	I2U0DS= "1"	At entry to control register setting wait state after end of data reception mode
Stop condition	I2U0MD4= "1"	I2U0SPS= "1"	At detecting stop-condition waveform.
Re-start condition, and then the master selects another slave.	I2U0MD3= "1"	I2U0RAS= "1"	At end of slave-address reception mode without coinciding slave-address after detected re-start condition with I2USAA= "1".
Detect stop-condition for another slave.	I2U0MD4= "1", I2U0MD1= "0"	I2U0SPS="1", I2U0AS= "1"	At detecting stop-condition waveform with I2U0SAA= "1".

Chapter 14 UART

14. UART

14.1 General Description

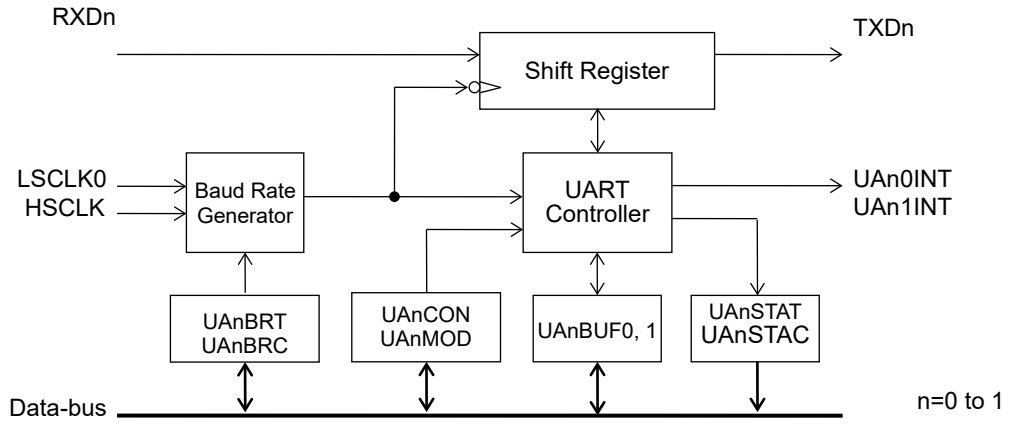
This LSI has full-duplexed universal asynchronous receiver transmitter; UART.

14.1.1 Features

- Data length : 5/6/7/8 bit
- Data parity : odd, even, fixed 0, fixed 1, none
- Stop bit : 1bit or 2bit
- Status flags : parity error, overrun error, framing error, transmission buffer
- Signal level : positive, negative
- Data direction : LSB first or MSB first
- Wide range of communication speed
  - 1bps to 4,800bps (Clock frequency is 32.768kHz)
  - 300bps to 2Mbps (Clock frequency is 16MHz)
- Built-in baud rate generator for each channel
- Half-duplex mode
- Self-test function using transmission and reception. See Chapter 29 "Safety Function." for the self-test functions.

14.1.2 Configuration

Figure 14-1 shows configuration of the UART.



UAnBUF0,1	:	UARTn transmission/reception buffer
UAnBRT	:	UARTn baud rate register
UAnBRC	:	UARTn baud rate adjustment register
UAnCON	:	UARTn control register
UAnMOD	:	UARTn mode register
UAnSTAT, UAnSTAC	:	UARTn status register, UARTn status clear register

Figure 14-1 Configuration of UART

14.1.3 List of Pins

The I/O pins of the UART are assigned to the shared function of the general ports.

Pin name	I/O	Description
RXDn	I	Reception data input of UART n
TXDn	O	Transmission data output of UART n

Table 14-1 shows the list of the general ports used for the UART and the register settings of the ports.

Table 14-1 Ports used for the UART and the register settings

●: available

Channel no.	Pin name	Shared port		Setting register	Setting value	Available/Unavailable	
						20 pin product	24 pin product
0	RXD0	P12*3	2nd Function	P1MOD2	0001_XXXX*1	●	●
	TXD0	P12*3	2nd Function	P1MOD2	0001_XXXX*2	●	●
		P13	2nd Function	P1MOD3	0001_XXXX*2	●	●
1	RXD1	P10*3	2nd Function	P1MOD0	0001_XXXX*1	●	●
	TXD1	P10*3	2nd Function	P1MOD0	0001_XXXX*2	●	●
		P23	2nd Function	P2MOD3	0001_XXXX*2	●	●

●: Available, -: Not available

*1 : "XXXX" determines the condition of the port input

XXXX	Condition of the port input
0001	Input (without an internal pull-up resistor)
0101	Input (with an internal pull-up resistor)

*2 : "XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	N-ch open drain output (without the pull-up)
1111	N-ch open drain output (with the pull-up)

*3 : Switching between TXDn and RXDn on P10 and P12 is set by the port function registers (P1MOD0, P1MOD2). When the port is 2nd function and output, it functions as a TXDn, and when it is 2nd function and an input, it functions as an RXDn.

14.2 Description of Registers

14.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF600	UART0 reception buffer	UA0BUF0	-	R	8	0x00
0xF601	UART0 transmission buffer	UA0BUF1	-	R/W	8	0x00
0xF602	UART0 status register	UA0STAT	-	R	8	0x00
0xF603	UART0 status clear register	UA0STAC	-	W	8	0x00
0xF604	UART0 control register	UA0CON	-	R/W	8	0x00
0xF605	Reserved register	-	-	-	-	-
0xF606	UART0 mode register	UA0MODL	UA0MOD	R/W	8/16	0x01
0xF607		UA0MODH		R/W	8	0x00
0xF608	UART0 interrupt enable register	UA0INTE	-	R/W	8	0x00
0xF609	Reserved register	-	-	-	-	-
0xF60A	UART0 baud rate register	UA0BRTL	UA0BRT	R/W	8/16	0xFF
0xF60B		UA0BRTH		R/W	8	0x0F
0xF60C	UART0 baud rate adjustment register	UA0BRC	-	R/W	8	0x00
0xF60D to 0xF60F	Reserved register	-	-	-	-	-
0xF610	UART1 reception buffer	UA1BUF0	-	R	8	0x00
0xF611	UART1 transmission buffer	UA1BUF1	-	R/W	8	0x00
0xF612	UART1 status register	UA1STAT	-	R	8	0x00
0xF613	UART1 status clear register	UA1STAC	-	W	8	0x00
0xF614	UART1 control register	UA1CON	-	R/W	8	0x00
0xF615	Reserved register	-	-	-	-	-
0xF616	UART1 mode register	UA1MODL	UA1MOD	R/W	8/16	0x01
0xF617		UA1MODH		R/W	8	0x00
0xF618	UART1 interrupt enable register	UA1INTE	-	R/W	8	0x00
0xF619	Reserved register	-	-	-	-	-
0xF61A	UART1 baud rate register	UA1BRTL	UA1BRT	R/W	8/16	0xFF
0xF61B		UA1BRTH		R/W	8	0x0F
0xF61C	UART1 baud rate adjustment register	UA1BRC	-	R/W	8	0x00

14.2.2 UARTn Reception Buffer (UAnBUF0)

UAnBUF0 is a SFR to store reception data.

Address : 0xF600(UA0BUF0),0xF610(UA1BUF0)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBUF0							
Bit	-	-	-	-	-	-	-	-	UnRD 7	UnRD 6	UnRD 5	UnRD 4	UnRD 3	UnRD 2	UnRD 1	UnRD 0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 0	UnRD7 to UnRD0	This works as the reception buffer. Data at the end of reception communication is overwritten into this register, so read out this register by using the UARTn0 interrupt generated at the end of reception communication. Writing to this register is invalid. When choosing the 5 to 7 bit length, unused bits return “0” for reading. (See to Section 14.3.2 “Data Direction”) In the transmission mode in half-duplex communication mode, UAnBUF0 is fixed at 0x00.

14.2.3 UARTn Transmission Buffer (UAnBUF1)

UAnBUF1 is a SFR to store transmission data.

Address : 0xF601(UA0BUF1),0xF611(UA1BUF1)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBUF1							
Bit	-	-	-	-	-	-	-	-	UnTD 7	UnTD 6	UnTD 5	UnTD 4	UnTD 3	UnTD 2	UnTD 1	UnTD 0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 0	UnTD7 to UnTD0	This works as the transmission buffer. Write transmission data to this. For continuous transmitting, write the next transmission data to this register after checking UnFUL bit of UAnSTAT is “0”. The written data in this register can be read out. When choosing the 5 to 7bit length, written data in unused bits are invalid. (See to Section 14.3.2 “Data Direction”)

14.2.4 UARTn Status Register (UAnSTAT)

UAnSTAT is a SFR to indicate states of the transmission/reception operation.

Address : 0xF602(UA0STAT),0xF612(UA1STAT)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnSTAT							
Bit	-	-	-	-	-	-	-	-	-	-	UnRXF	UnTXF	UnFUL	UnPER	UnOER	UnFER
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 6	-	Reserved bits
5	UnRXF	Indicate reception state. 0: Data reception is stopped (Initial value) 1: Data reception is in progress
4	UnTXF	Indicate transmission state. 0: Data transmission is stopped (Initial value) 1: Data transmission is in progress
3	UnFUL	Indicate transmission buffer state. This bit is set to “1” by writing a data to UAnBUF1 and reset to “0” when starting to transfer the data. To transmit data successively, check that the UnFUL bit is "0" before writing the next transmission data to the UAnBUF1.This bit is forcibly reset to "0" by writing “1” to the UnFULC bit of UAnSTAC register. 0: Transmission buffer has no data (Initial value) 1: Transmission buffer has data
2	UnPER	Indicate a parity error. The parity of the received data and the parity bit added to the data are compared and if they do not match, this bit becomes "1".This bit is forcibly reset to "0" by writing “1” to the UnPERC bit of UAnSTAC register. 0: The parity error has not occurred. (Initial value) 1: The parity error has occurred.
1	UnOER	Indicate a reception overrun error. This bit becomes "1" if the next data is received before reading the previous receive data in reception buffer (SDnBUFL). Even if reception is stopped by the UnEN bit and then reception is re-started, this bit is set to “1” unless the previously received data is not read. Therefore, make sure that data is always read from the reception buffer even if the data is not required. This is forcibly reset to "0" by writing “1” to the UnOERC bit of UAnSTAC register. 0: The overrun error has not occurred (Initial value) 1: The overrun error has occurred
0	UnFER	Indicate a framing error. This becomes "1" when an error occurs in the start/stop bit. The Un0FER bit is forcibly reset to "0" by writing “1” to the UnFERC bit of UAnSTARC register. 0: The framing error has not occurred (Initial value) 1: The framing error has occurred

14.2.5 UARTn Status Clear Register (UAnSTAC)

UAnSTAC is a write-only SFR to clear states of the transmission/reception operation.

Address : 0xF603(UA0STAC),0xF613(UA1STAC)
Access : W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnSTAC							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	UnFULC	UnPERC	UnOERC	UnFERC
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 4	-	Reserved bits
3	UnFULC	Clear UnFUL bit ; transmission buffer state flag. Writing “0” : Invalid Writing “1” : Clear the UnFUL bit
2	UnPERC	Clear UnPER bit ; parity error flag. Writing “0” : Invalid Writing “1” : Clear the UnPER bit.
1	UnOERC	Clear UnOER bit ; overrun error flag. Writing “0” : Invalid Writing “1” : Clear the UnOER bit.
0	UnFERC	Clear UnFER bit ; framing error flag. Writing “0” : Invalid Writing “1” : Clear the UnFER bit.

14.2.6 UARTn Control Register (UAnCON)

UAnCON is a SFR to control enable/disable communication.

Address : 0xF604(UA0CON),0xF614(UA1CON)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UnEN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	UnEN	Enable the UART n communication. See section 14.3.4 “Transmission Operation” for details. A both of transmission and reception is enabled when UnEN = “1”. Set “0” to this bit if communication is stopped. 0: Disabled (Initial value) 1: Enabled

[Note]

- **Do setting for used ports and the mode/baud rate before setting “1” to UnEN bit.**

14.2.7 UARTn Mode Register (UAnMOD)

UAnMOD is a SFR to set the transfer mode.

Address : 0xF606(UA0MODL/ UA0MOD), 0xF607(UA0MODH),
0xF616(UA1MODL/UA1MOD), 0xF617(UA1MODH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0001

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAnMOD															
Byte	UAnMODH								UAnMODL							
Bit	UnDIR	UnNEG	UnSTP	UnPT2	UnPT1	UnPT0	UnLG1	UnLG0	UnHD	UnRSS	-	-	-	-	UnCK0	UnIO
R/W	R/W	R/W	R/W	R/W	R/w	R/w	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit no.	Bit symbol name	Description
15	UnDIR	Select the data direction. 0: LSB first (Initial value) 1: MSB first
14	UnNEG	Select a logic of the data input / output. 0: Positive logic (Initial value) 1: Negative logic
13	UnSTP	Select a stop bit length. 0: 1 stop bit (Initial value) 1: 2 stop bit
12 to 10	UnPT2 to UnPT0	Select a parity bit. XX0: No parity bit (Initial value) 001: Odd parity 011: Even parity 101: Parity bit is fixed to “1” 111: Parity bit is fixed to “0” X: 0 or 1 (don’t care)
9 to 8	UnLG1 to UnLG0	Select a data length. 00: 8-bit length (Initial value) 01: 7-bit length 10: 6-bit length 11: 5-bit length
7	UnHD	Select the communication mode of the UARTn. 0: Full-duplex communication mode (Initial value) 1: Half-duplex communication mode
6	UnRSS	Select sampling timing of the reception data. 0: (Values set to UAn0BRTH and UAn0BRTL registers)/2 (Initial value) 1: {(Values set to UAn0BRTH and UAn0BRTL registers)/2} -1
5 to 2	-	Reserved bits
1	UnCK0	Select base clock baud rate generator. 0: LSCLK0 (Initial value) 1: HSCLK
0	UnIO	Select the transmission/reception in the half-duplex communication mode of UARTn. When full-duplex communication mode is selected, the setting is disabled. 0: Transmission mode 1: Receive mode (Initial value)

[Note]

- **Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0).**

14.2.8 UARTn Interrupt Enable Register (UAnINTE)

UAnINTE is a SFR to enable interrupt requests.

Address : 0xF608 (UA0INTE), 0xF618 (UA1INTE)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnINTE							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	UnFIE	UnTIE	UnRIE
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits:
It is configured enable/disable a target interrupt
0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

Bit no.	Bit symbol name	Description
7 to 3	-	Reserved bits
2	UnFIE	Transmission completion interrupt This has occurred when transmission data is transmitted in the condition of transmission buffer empty.
1	UnTIE	Transmission buffer empty interrupt. This has occurred when transmission buffer becomes empty.
0	UnRIE	Reception interrupt. This has occurred at receiving a data.

14.2.9 UARTn Baud Rate Register (UAnBRT)

UAnBRT is a SFR to set the count value of the baud rate generator in UARTn.
For details of relation between the count value of the baud rate generator and the baud rate, see Section 14.3.3 “Baud Rate”.

Address : 0xF60A (UA0BRTL/UA0BRT), 0xF60B (UA0BRTH),
0xF61A (UA1BRTL/UA1BRT), 0xF61B (UA1BRTH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0FFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAnBRT															
Byte	UAnBRTH								UAnBRTL							
Bit	UnBR 15	UnBR 14	UnBR 13	UnBR 12	UnBR 11	UnBR 10	UnBR 9	UnBR 8	UnBR 7	UnBR 6	UnBR 5	UnBR 4	UnBR 3	UnBR 2	UnBR 1	UnBR 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

14.2.10 UARTn Baud Rate Adjustment Register (UAnBRC)

UAnBRC is a SFR to adjust the count value of the baud rate generator in UARTn.
For details of relation between the value of UAnBRC and the correction value, see Section 14.3.3 “Baud Rate”.

Address : 0xF60C(UA0BRC),0xF61C(UA1BRC)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBRC							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	UnBR C2	UnBR C1	UnBR C0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Be sure to set UAnBRT and UAnBRC register while communication is stopped (UnEN=0). Do not rewrite it during communication.**

14.3 Description of Operation

14.3.1 Frame Format

In the transfer data format, one frame contains a start bit, a data bit, a parity bit, and a stop bit. In this format, the following are choosable: 5 to 8 bits for the data bit, even/odd/ fixed to "1", or fixed to "0" for the parity bit, 1 stop bit or 2 stop bit for the stop bit, LSB first or MSB first for the transfer direction, and positive logic or negative logic for the logic of the serial input/output.
All of these are set in the UARTn mode register (UAnMOD).
Figure 14-2 shows the input/output format.

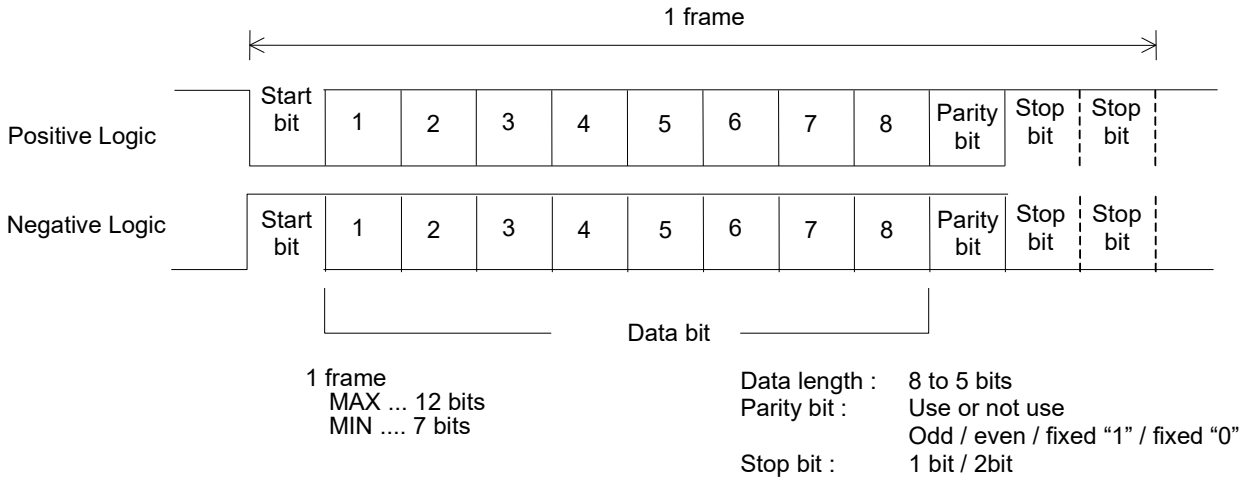
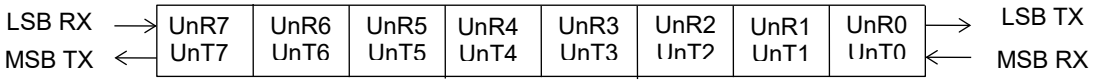


Figure 14-2 Format of Input/Output (LSB first)

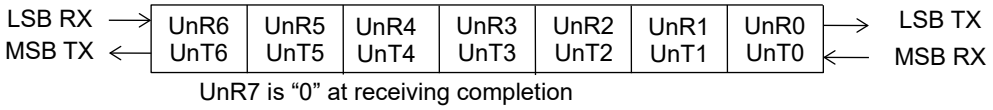
14.3.2 Data Direction

Figure 14-3 shows a relationship between the transmission/reception buffer and data.

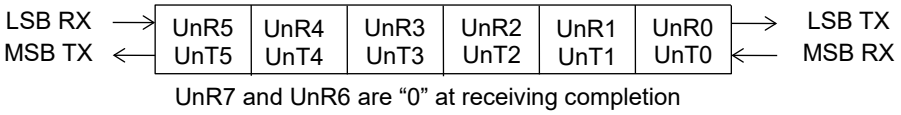
- 8-bit length data



- 7-bit length data



- 6-bit length data



- 5-bit length data

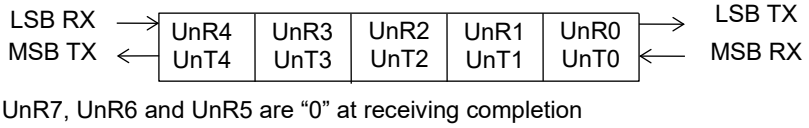


Figure 14-3 Relationship between the Transmission/Reception Buffer and Data

14.3.3 Baud Rate

The baud rate generator generates a baud rate using the base clock chosen in the UARTn mode register (UAnMOD). The setting values for the UARTn baud rate register (UAnBRT) and the UARTn baud rate adjustment register (UAnBRC) can be calculated by the following formulae.

UAnBRT = ROUNDDOWN (Base clock frequency (Hz) / Baud rate (bps)) – 1 + Carryover of UAnBRC
UAnBRC = ROUND ((Base clock frequency (Hz) % Baud rate (bps)) × 8 / Baud rate (bps))

where is ROUNDDOWN: Rounded down, ROUND: Rounded to the nearest whole number, %:Surplus.
Setting range of UAnBRC is 0 to 7. If the calculated value of UAnBRC is 8, add 1 to UAnBRT and set 0 to UAnBRC.

Example(1) : Base clock frequency: Approx.16 MHz (16.003072MHz), Baud rate ideal value: 115,200 bps
16.003072MHz / 115,200bps - 1 = 138.91555⋯ - 1 = 137 (rounding down to the nearest integer)
= 0x0089
(16.003072MHz % 115,200bps) × 8 / 115,200bps = 105472 × 8 / 115,200
= 7.32444⋯ = 7 (rounding to the nearest integer)
= 0x07
UAnBRT = 0x0089, UAnBRC = 0x07

Example(2) : Base clock frequency: Approx.16MHz (16.003072MHz), Baud rate ideal value:4,800bps
16.003072MHz / 4,800bps - 1 = 3333.97333⋯ - 1 = 3332 (rounding down to the nearest integer)
= 0x0D04
(16.003072MHz % 4,800bps) × 8 / 4,800bps = 4672 × 8 / 4,800
= 7.78666⋯ = 8 (rounding to the nearest integer)
= 0x08 UAnBRC carrier over occurred
UAnBRT = 0x0D04+1 = 0x0D05, UAnBRC = 0x08 = 0x00

The actual baud rate calculated from the setting value for the baud rate can be expressed by the following formula:

Actual baud rate (bps) = [Base clock frequency] / {(UAnBRT + 1) + (UAnBRC / 8)}

Example: Base clock frequency: Approx.16 MHz (16.003072 MHz), Baud rate ideal value: 1200 bps
Actual baud rate (bps) = 16.003072MHz / {(0x3416 + 1)} + (0x07 / 8)} ≈1200.00

Table 14-2 lists the count values for typical baud rates.

Table 14-2 Count Values for Typical Baud Rates

Base clock	Baud rate	UAnBRT	UAnBRC	Actual baud rate
HSCLK (Approx. 16.003072MHz)	300bps	0xD05E	0x05	300.00bps
	1,200bps	0x3416	0x07	1200.00bps
	2,400bps	0x1A0B	0x00	2399.98bps
	4,800bps	0x0D05	0x00	4799.96bps
	9,600bps	0x0682	0x00	9599.92bps
	19,200bps	0x0340	0x04	19199.85bps
	38,400bps	0x019F	0x06	38399.69bps
	57,600bps	0x0114	0x07	57590.90bps
	115,200bps	0x0089	0x07	115233.64bps
LSCLK (32.768kHz)	200bps	0x00A2	0x07	199.95bps
	300bps	0x006C	0x02	299.93bps
	1,200bps	0x001A	0x02	1202.49bps
	2,400bps	0x000C	0x05	2404.99bps
	4,800bps	0x0005	0x07	4766.25bps

14.3.4 Transmission Operation(Common to full-duplex and half-duplex communication modes)

In full-duplex communication mode, set the UnHD bit of the UARTn mode register (UAnMOD) to "0" to select full-duplex communication mode.
Transmission is started by setting the UnEN bit of the UART n control register (UAnCON) to "1" and set transfer data to UAnBUF1. The order of UAnEN setting and UAnBUF1 setting does not matter.
Figure 14-4 shows the operation timing for transmission.

When the UnEN bit is set to "1", the transmission status; UnTXF is set to “1” after 2 cycles of the system clock. (at (1))
An internal transfer clock of baud rate supplies after 2 cycles of the base clock (LSCLK0/HSCLK), and then the start bit is output the TXD pin. (at (2)) Subsequently, the transmitted data, a parity bit, and a stop bit are output.
When the start bit is output, the transmission buffer status flag; UnFUL is return to “0” and the transmission interrupt is requested on the rising edge of the internal transfer clock. (at (3)) In the UARTn transmission interrupt routine, the next data to be transmitted is written to the transmission buffer (UAnBUF1). Then the UnFUL is set to "1". It is same as (2) after transmission of the stop bit (at (4)). At this time if the UART transmission interrupt routine is terminated without writing the next data to the transmit buffer; it means the stop-bit is sent when UnFUL is not set to “1”, transmission is stop. Then the UnTXF bit is reset to "0", and the UART transmission interrupt is requested. (at (5))
The valid period for the next transmission data to be written to the transmission buffer is from when the UnFUL bit becomes “0” after the interrupt occurs to the termination of stop bit transmission. (at (6))

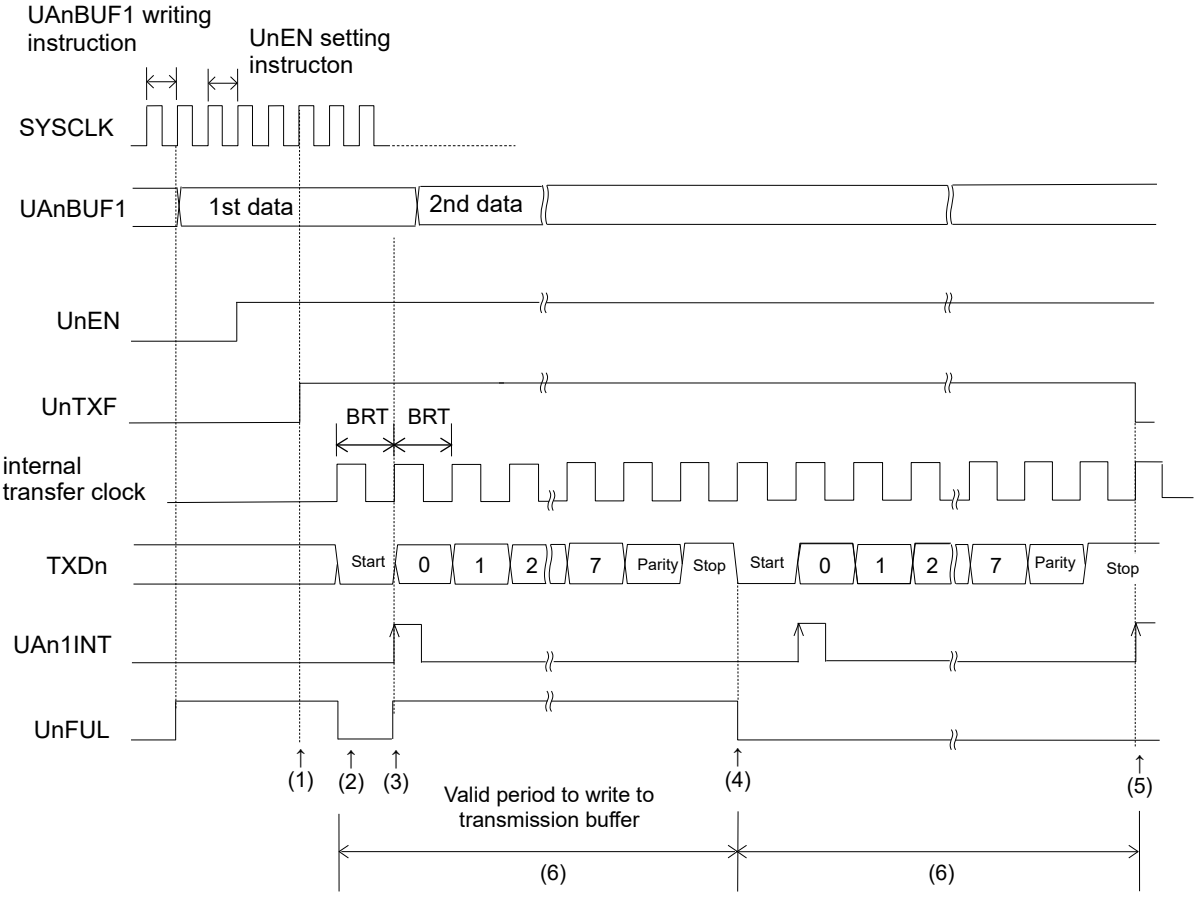


Figure 14-4 Transmission Timing in full-duplex mode

[Note]

- **The transmission is start when setting “1” to UnEN bit of UAnCON with UnFUL bit =1. Write “1” to UnFULC bit in UAnSTAC register to reset UnFUL bit, and then set “1” to UnEN to allow transmission/reception, if the transmission data is not ready and the reception is permitted first.**

14.3.5 Reception Operation (Common to full-duplex and half-duplex communication modes)

In full-duplex communication mode, set the UnHD bit of the UARTn mode register (UAnMOD) to "0" to select full-duplex communication mode.
Reception is started by setting the UnEN bit of the UARTn control register (UAnCON) to "1".
In the case of half-duplex communication mode reception, set the UnHD bit of the UARTn mode register (UAnMOD) to "1" to select the half-duplex communication mode, and set the UnIO bit of the UARTn mode register (UAnMOD) to "1" to select the reception mode.
A reception is started by setting the UnEN bit of the UART n control register (UAnCON) to "1".
Figure 14-5 shows the operation timing for reception.

When a reception starts, this module checks the data sent to the input pin RXD and waits for the arrival of a start bit.
When detecting a start bit ((2) in Figure 14-6), It generates the internal transfer clock of the baud rate set with the start bit detect point as a reference and performs reception operation.
The shift register shifts in the data input to RXD on the rising edge of the internal transfer clock. The data and parity bit are shifted into the shift register and 5- to 8- bit received data is transferred to the reception buffer (UAnBUF0) concurrently with the falling edge of the internal transfer clock of (3) in Figure 14-6.
This module requests a UART reception interrupt on the rising edge of the internal transfer clock subsequent to the internal transfer clock by which the received data was fetched ((4) in Figure 14-6) and checks for a stop bit error and a parity bit error. When an error is detected, this module sets the corresponding bit of the UART n status register (UAnSTAT) to "1".

Parity error : UnPER ="1"
Overrun error : UnOER ="1"
Framing error : UnFER ="1"

As shown in Figure 14-5, the rise of the internal transfer clock is set so that it may fall into the middle of the bit interval of the received data.
A reception continues until the UnEN bit is reset to "0" by the program. When the UnEN bit is reset to "0" during reception, the received data may be destroyed. When the UnEN bit is reset to "0" during the "UnEN reset enable period" in Figure 14-5, the received data is protected.

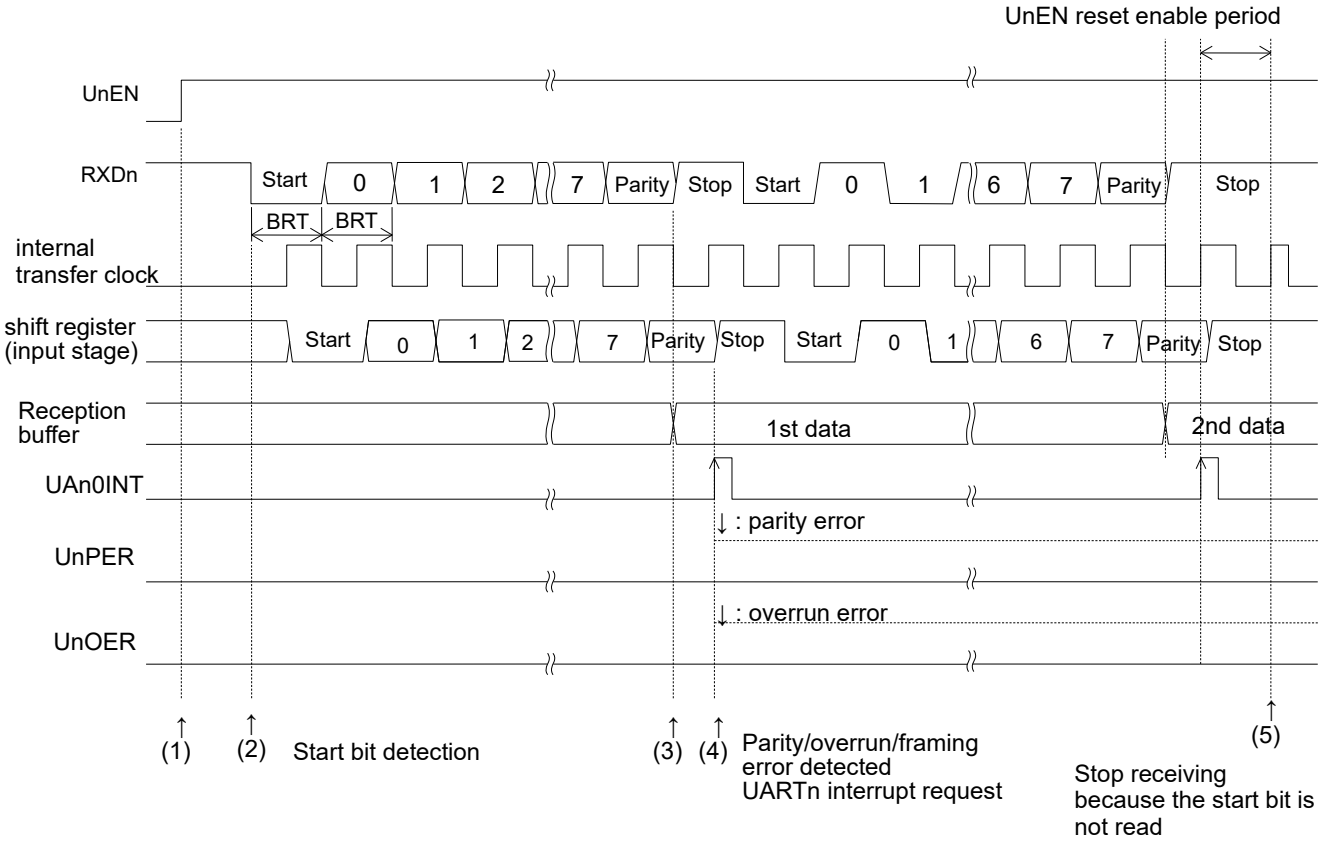


Figure 14-5 Reception Timing

14.3.5.1 Detection of Start Bit

The start bit is sampled with the baud rate generator clock selected by UnCK0 bit of UAnMOD register. Therefore, the start bit detection may be delayed for one cycle of the baud rate generator clock at the maximum.
Figure 14-6 shows the start bit detection timing.

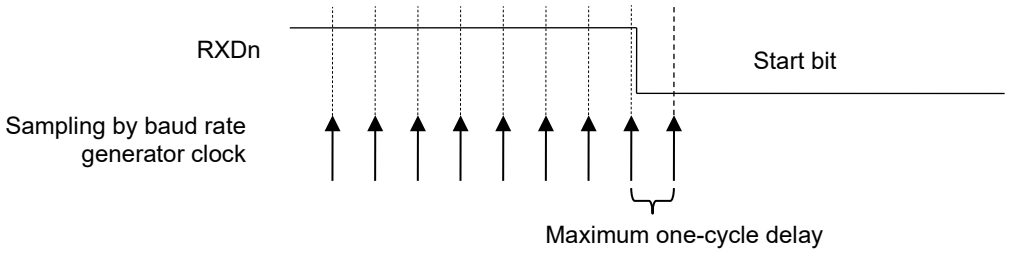
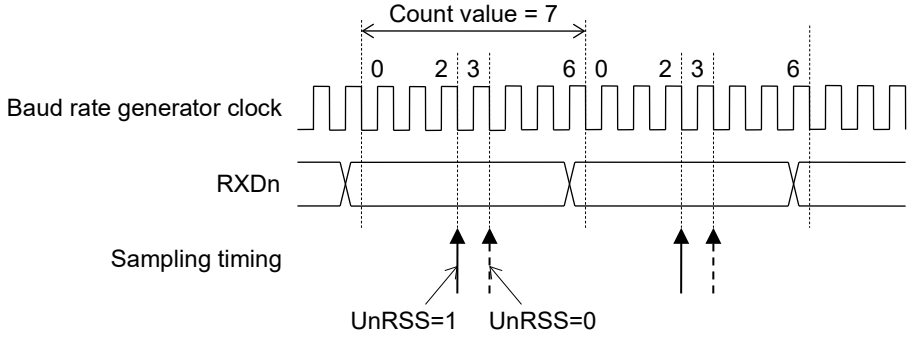


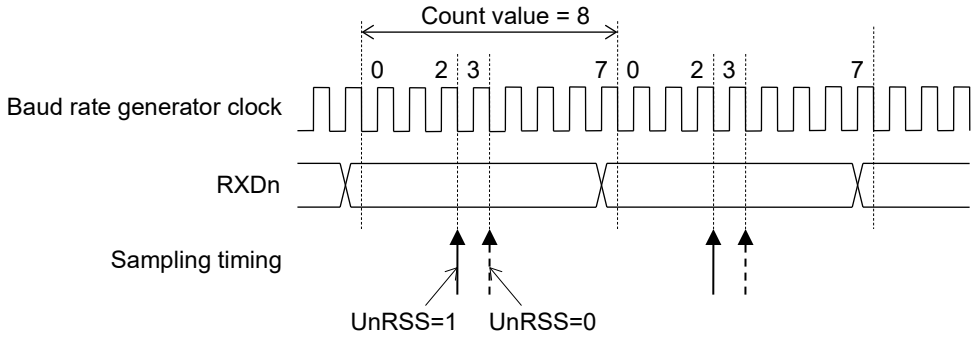
Figure 14-6 Start Bit Detection Timing (with Positive Logic)

14.3.5.2 Sampling Timing

When the start bit is detected, the received data that was input to RXDn is sampled almost at the center of the baud rate, and then loaded to the shift register.
This sampling timing the shift register uses to load data can be adjusted for one clock of the baud rate generator clock in UnRSS bit of the UART n mode register (UAnMOD).
Figure 14-7 shows the relationship between UnRSS bit and the sampling timing.



(1) When the baud rate generator count value is "7" (odd)



(2) When the baud rate generator count value is "8" (even)

Figure 14-7 Relationship between UnRSS Bit and Sampling Timing

14.3.5.3 Receiving Margin

If there is an error between the sender baud rate and the receiver baud rate generated by the baud rate generator, the error accumulates until the last stop bit loading in one frame, decreasing the reception margin.
Figure 14-8 shows the baud rate errors and reception margin waveforms.

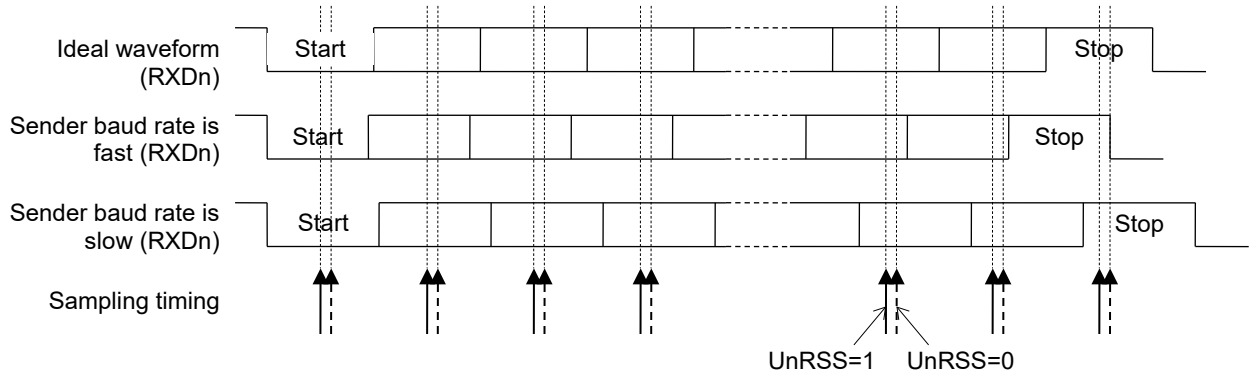


Figure 14-8 Baud Rate Errors and Reception Margin

[Note]

- **When designing the system, consider the difference of the baud rate between the transmission side and reception side, a delay of the start bit detection, signal degradation and noise influence, then adjust the baud rate and reception timing to ensure sufficient receiving margin.**

14.3.5.4 Reception Filter

This unit has reception data filter for a noise reduction. Figure 14-9 shows the RXD0 waveform before/after noise reduction.

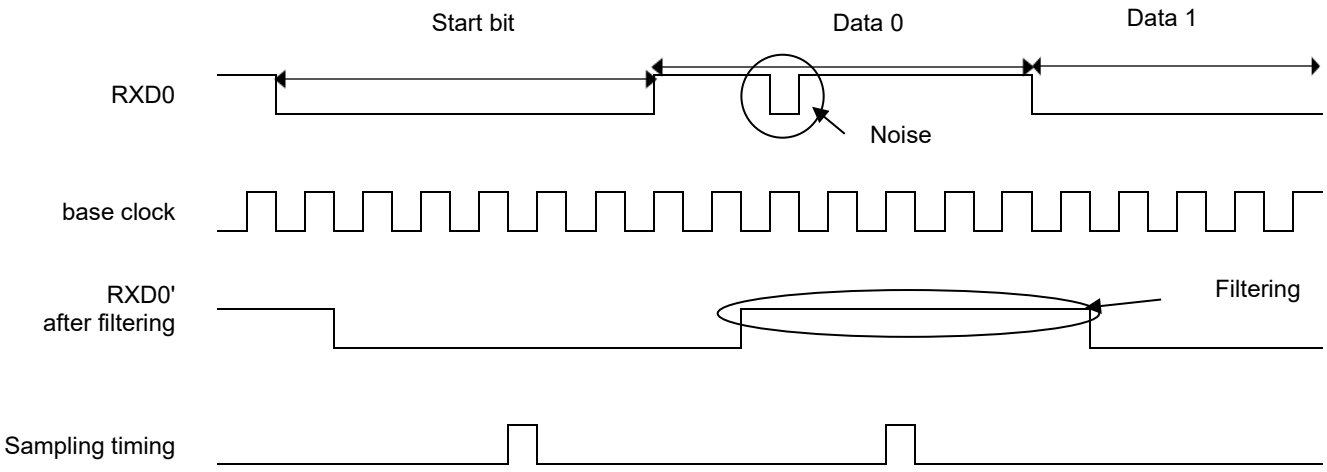


Figure 14-9 Noise reduction

14.3.6 Interrupt

Figure 14-10 shows the interrupt timing.
The transmission empty interrupt is generated as UAn1INT at the end of start-bit after a transmission buffer becomes empty. At this time, UnFUL=0 and UnTXF=1.
The transmission completion interrupt is generated as UAn1INT when a transmission is completed in condition of that the transmission buffer is empty. At this time, UnFUL=0, UnTXF=0.
The reception interrupt is generated as UAn0INT when reception data is stacked the buffer.

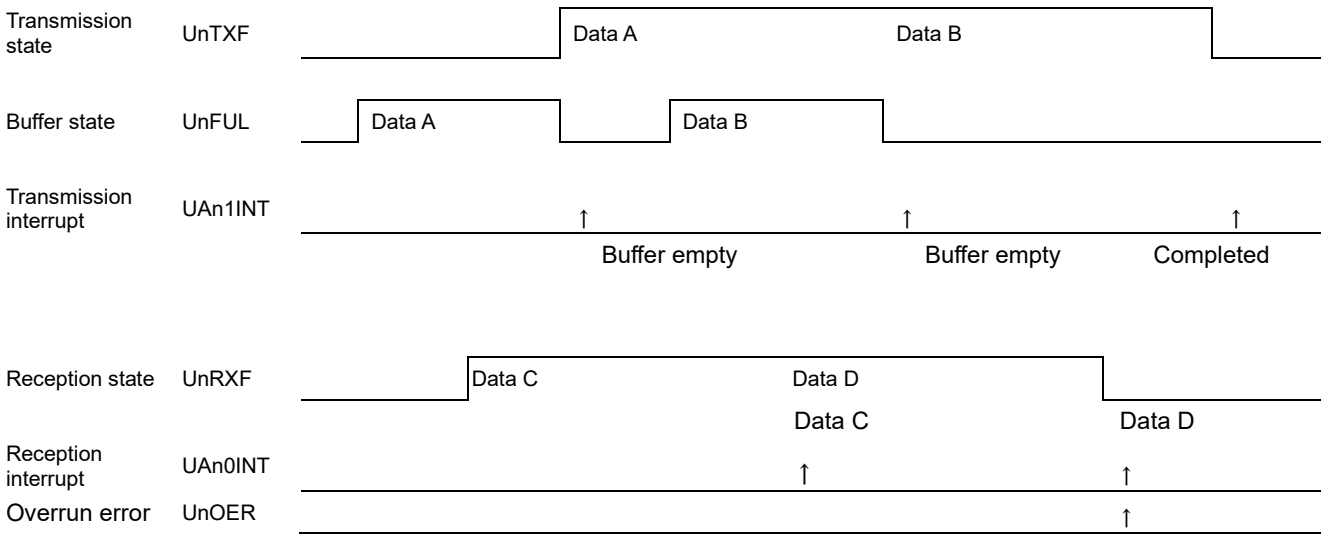


Figure 14-10 Interrupt Timing

14.3.7 Example of Setting

Figure 14-11 shows the example of the setting transmission operation in full-duplex communication mode using UART.

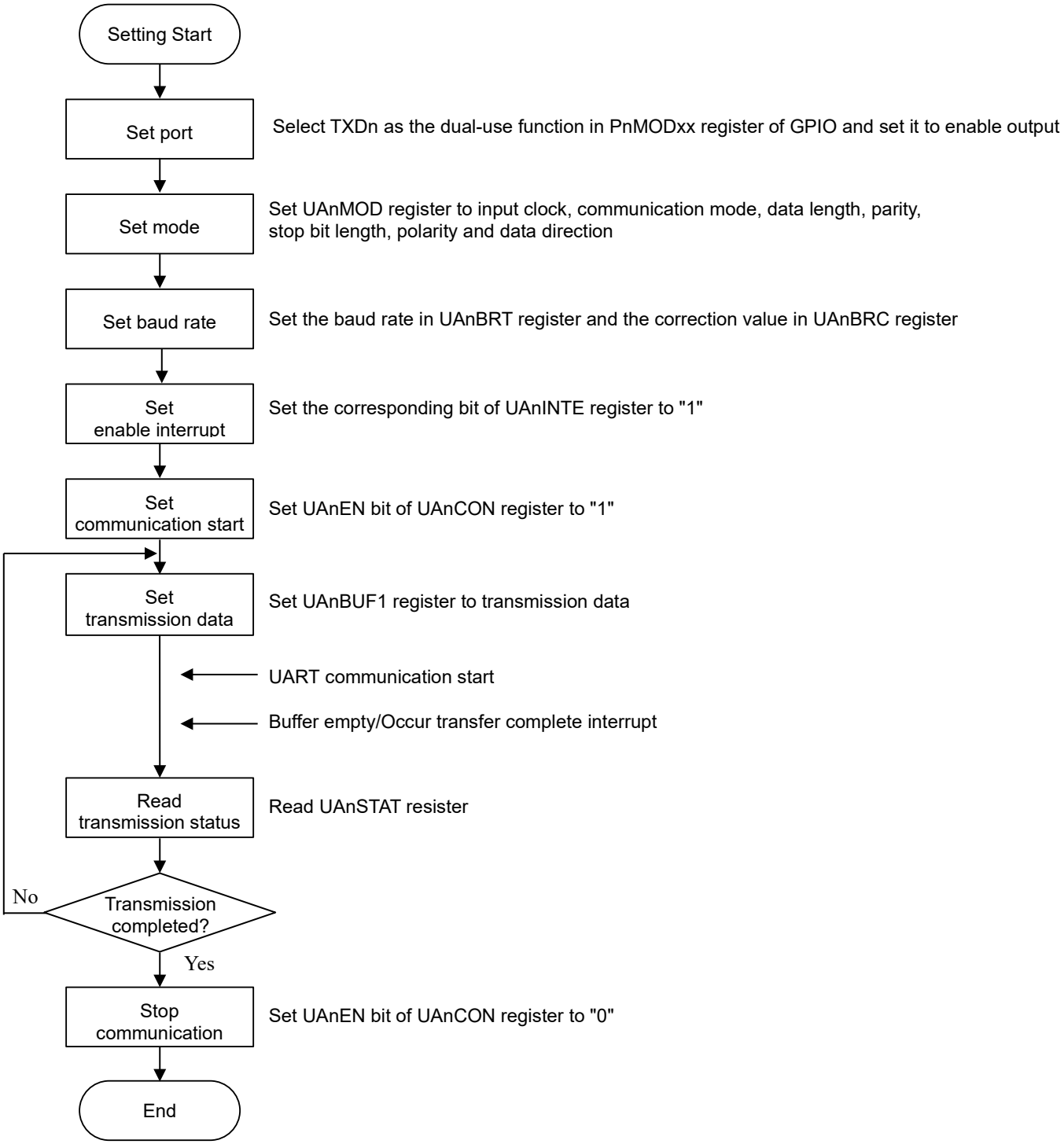


Figure 14-11 Example of transmission operation in full-duplex communication mode using UART

Figure 14-12 shows an example of the receive operation in full-duplex communication mode using UART.

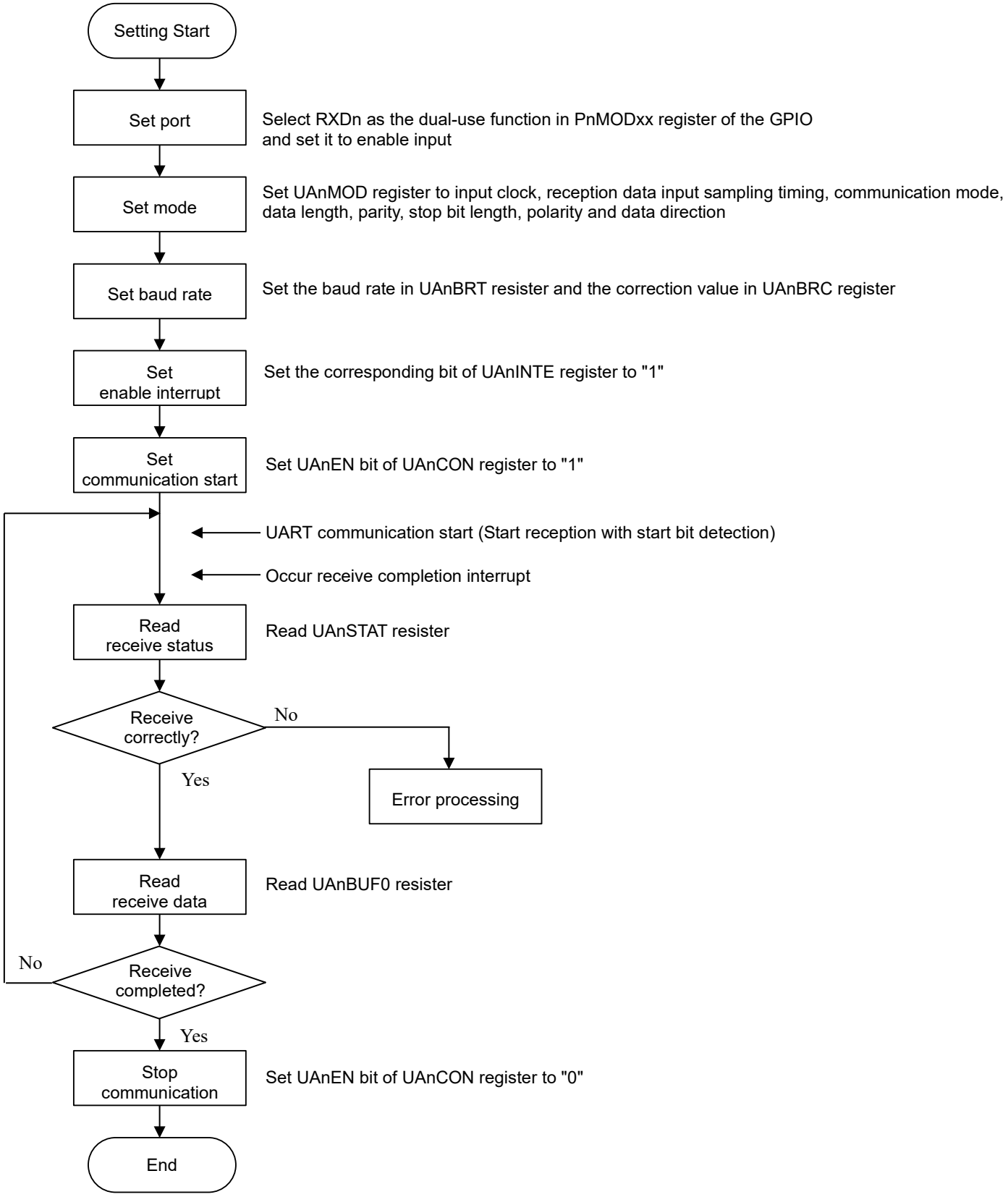
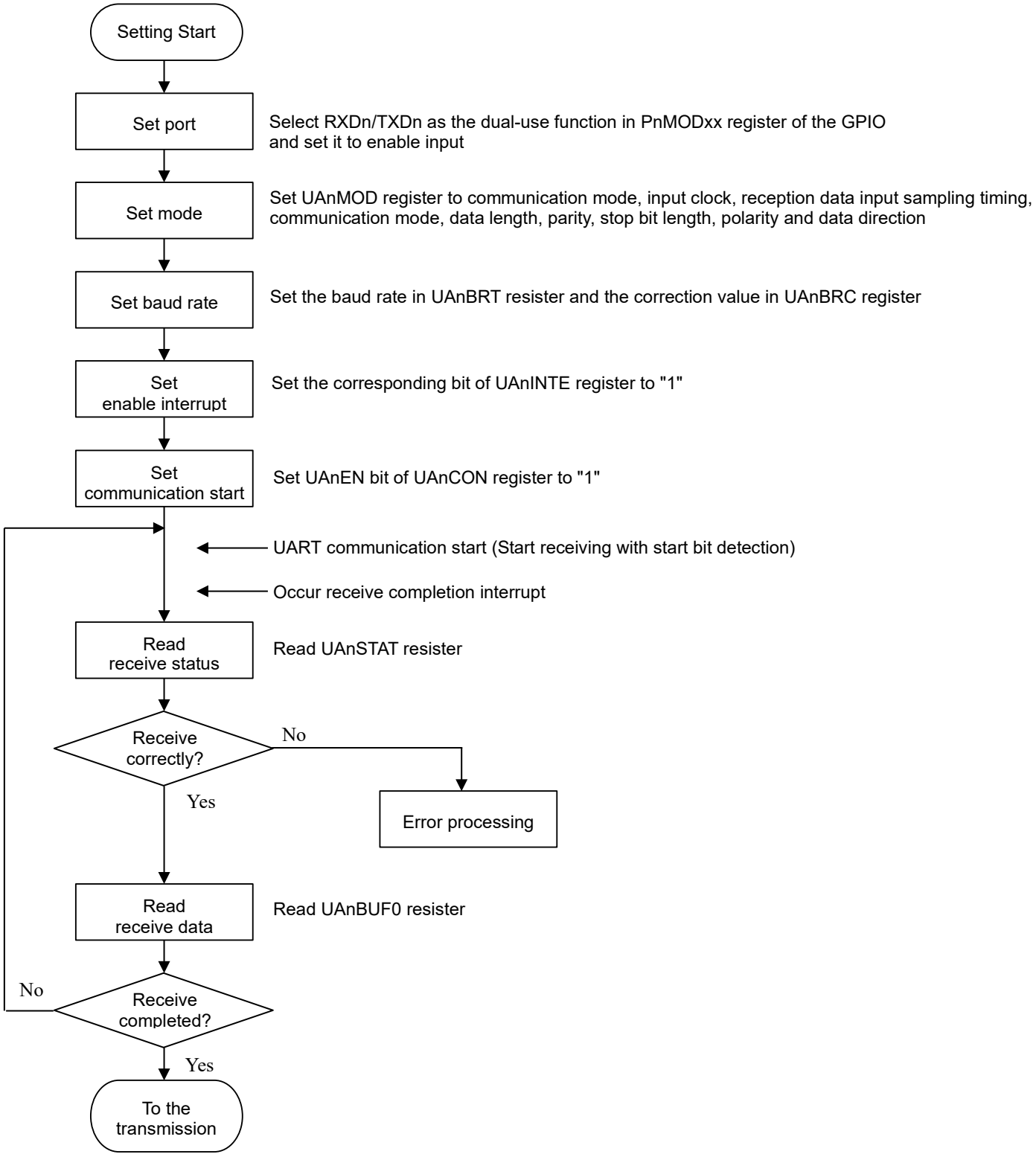
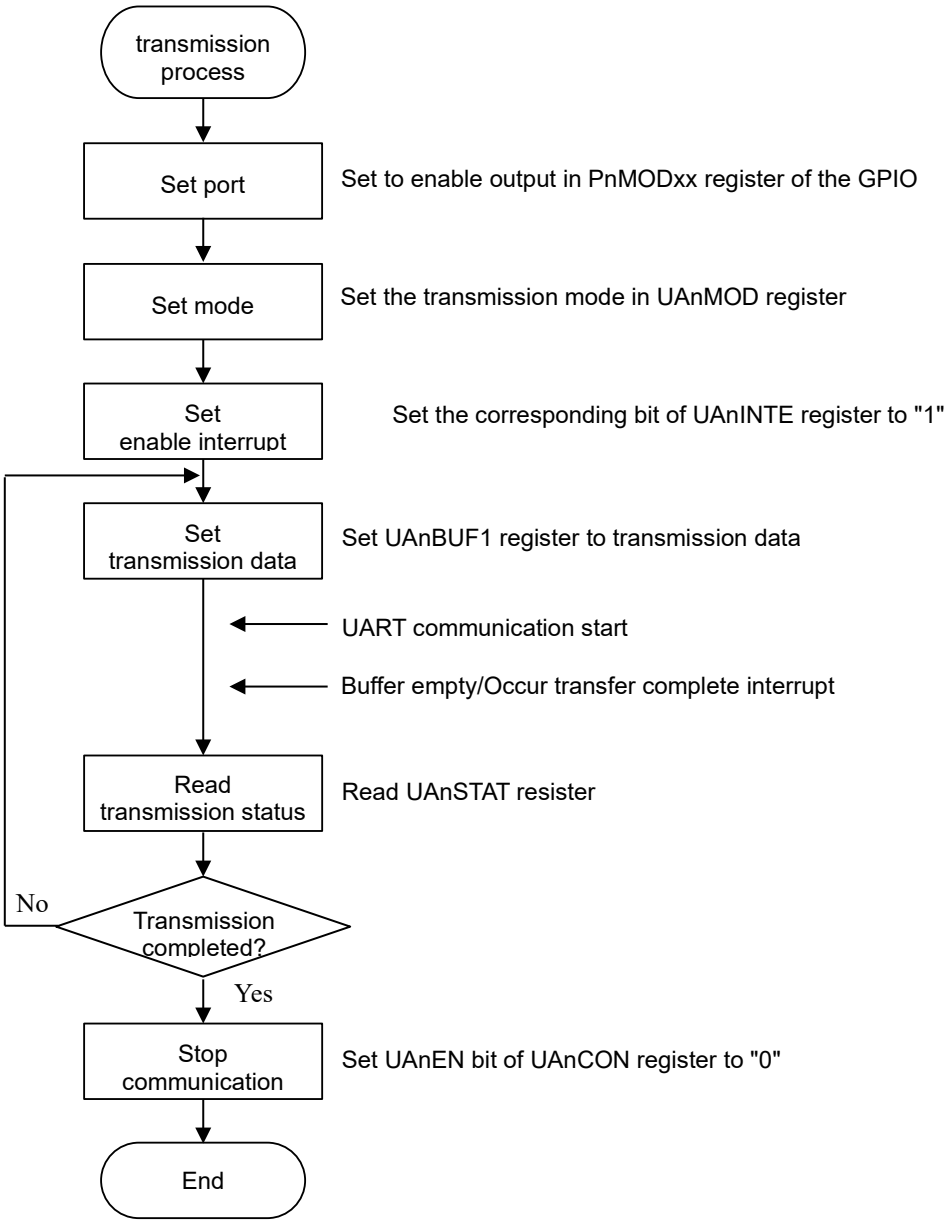


Figure 14-12 Example of receive operation in full-duplex communication mode using UART

Figure 14-13 shows an example of setting for half-duplex communication mode (receive operation → transmission operation) using UART.



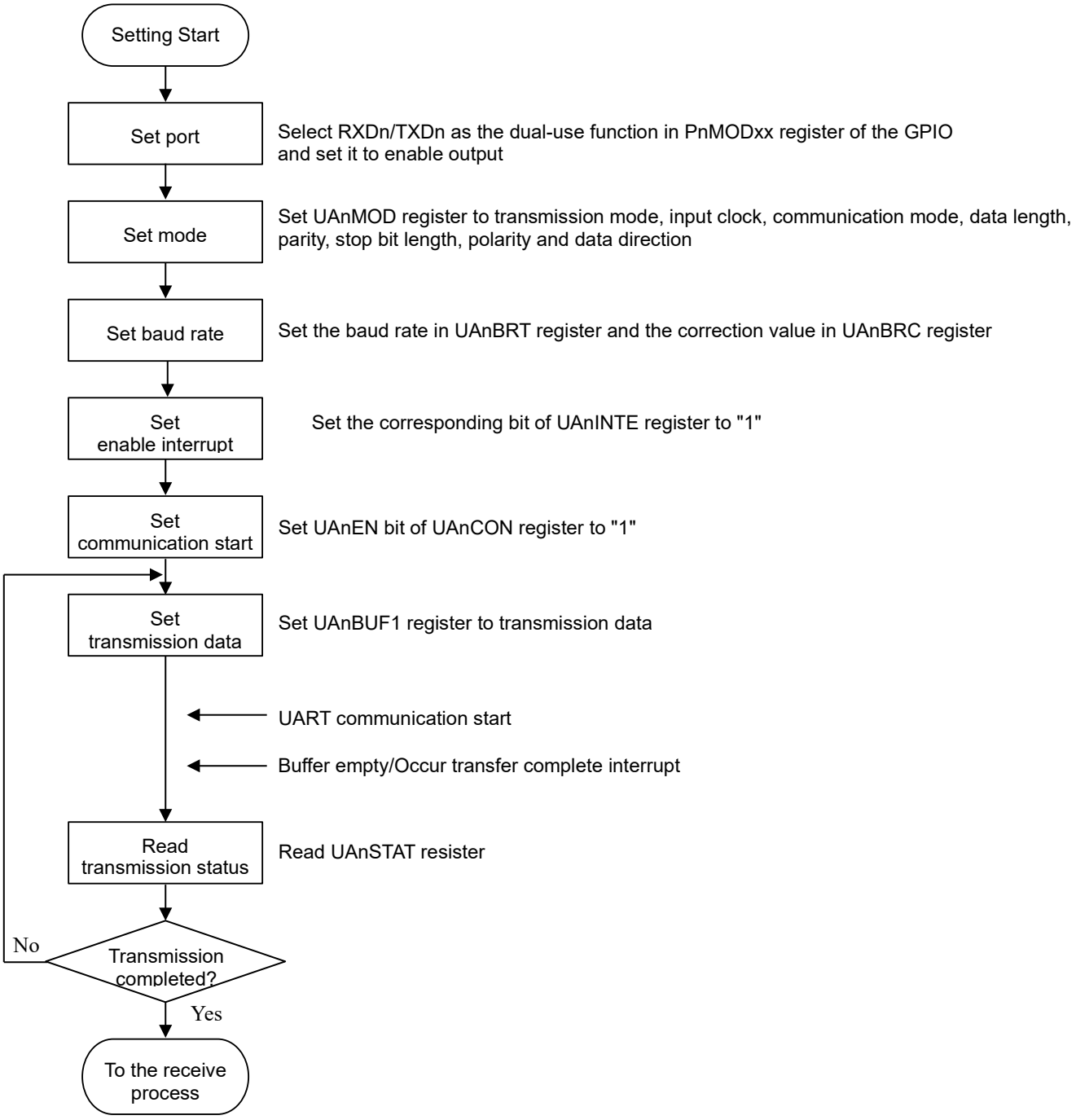
(a) Example of setting receive during (receive operation → transmission operation)



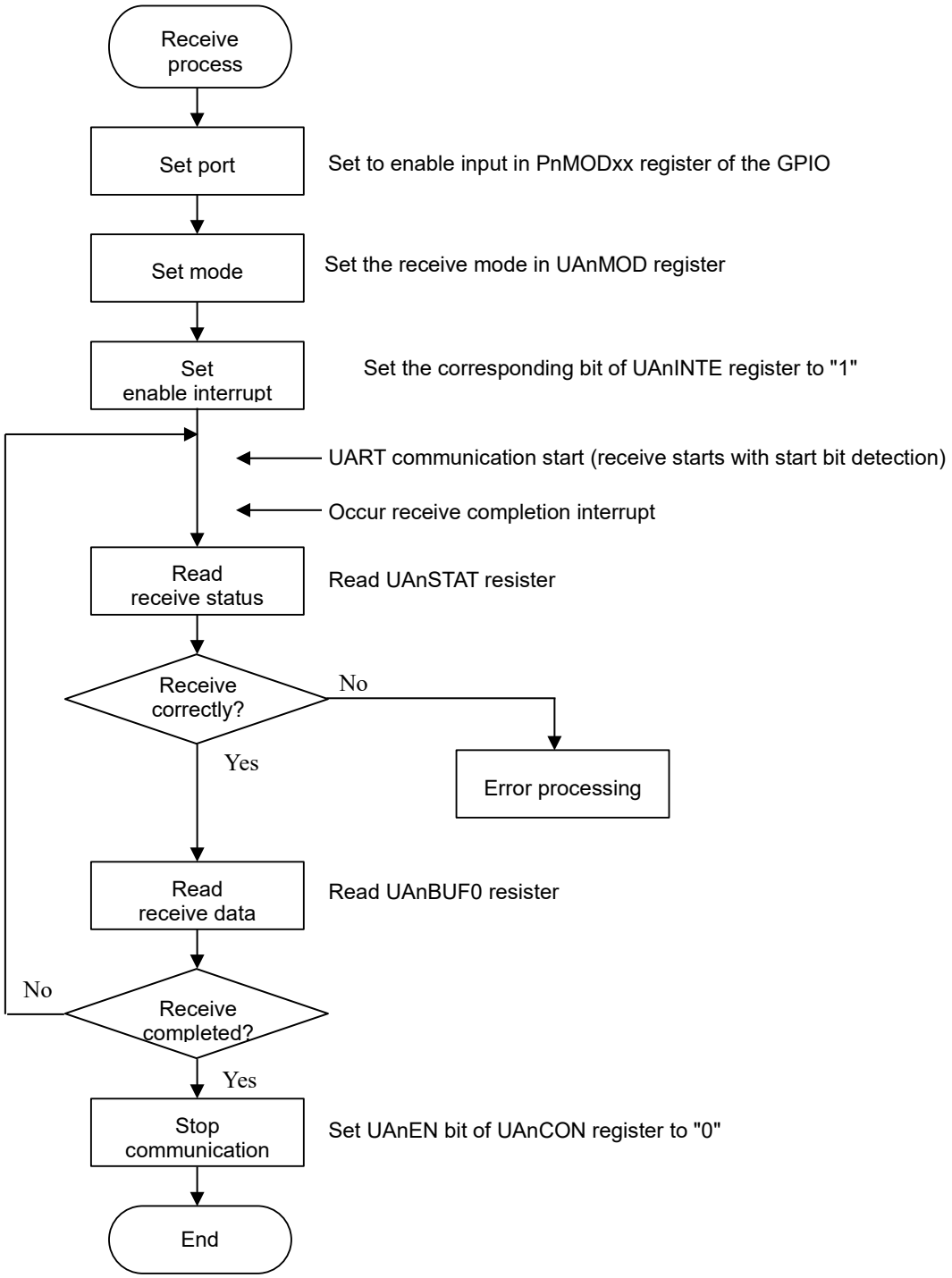
(b) Example of setting transmission during (receive operation → transmission operation)

Figure 14-13 Example of half-duplex communication mode (receive operation → transmission operation) using UART

Figure 14-14 shows an example of setting for half-duplex communication mode (transmission operation → receive operation) using UART.



(a)Example of setting transmission during (receive operation → transmission operation)



(b)Example of setting receive during (receive operation → transmission operation)

Figure 14-14 Example of half-duplex communication mode (transmission operation → receive operation) using UART

Chapter 17 GPIO

17. GPIO

17.1 General Description

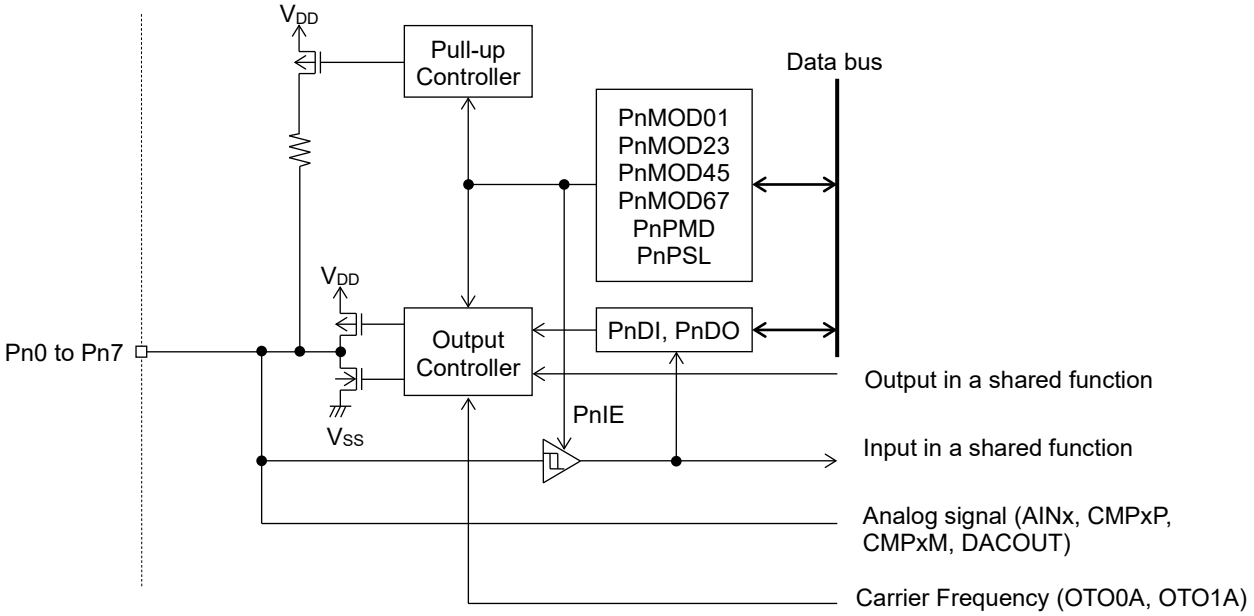
The input and output of a GPI/O is switchable on each pin. Max. 8 pins are available to read or to change the level of output in the same time. A general input port or output port shares a number of functions. See to Section 1.3.2 " List of Pins" or 1.3.3 "Description of Pins" for more detail.
The number of general port is dependent of each product. See Table 17-1 "List of Pins".

17.1.1 Features

- Selectable input/output function in each pin
- Selectable with or without Pull-up resistor in each pin
- Selectable CMOS output or N-channel open drain output or P-channel open drain output in each pin
- Direct driving LEDs when the N-channel open drain output is selected
- Carrier frequency output function
- Port output level test function

17.1.2 Configuration

Figure 17-1 shows the configuration of the general purpose port. See to Section 17.2.1 "List of registers" for available pins and registers.



PnDI : Port n data register (bit 7 to 0)
PnDO : Port n data register (bit 15 to 8)
PnMOD01 : Port n mode register 01
PnMOD23 : Port n mode register 23
PnMOD45 : Port n mode register 45
PnMOD67 : Port n mode register 67
PnPMD : Port n pulse mode register
PnPSL : Port n pulse selection register

Figure 17-1 Configuration of GPI/O port n

17.1.3 List of Pins

Table 17-1 List of Pins (● is available)

Pin Name	Primary Function	Available/Unavailable	
		20 pin product	24 pin product
P00/TEST0	GPI/O	●	●
P01	GPI/O	●	●
P02	GPI/O	●	●
P03	GPI/O	●	●
P04	GPI/O	●	●
P05	GPI/O	●	●
P06	GPI/O	●	●
P07	GPI/O	-	●
P10	GPI/O	●	●
P11	GPI/O	●	●
P12	GPI/O	●	●
P13	GPI/O	●	●
P14	GPI/O	●	●
P15	GPI/O	●	●
P16	GPI/O	●	●
P17	GPI/O	●	●
P20	GPI/O	-	●
P21	GPI/O	-	●
P22	GPI/O	-	●
P23	GPI/O	●	●

17.2 Description of Registers

17.2.1 List of Registers

Writing to SFRs of unequipped port is not available. PnDI return 0xFF for reading. Other SFRs return 0x0000/0x00 for reading.

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF200	Port 0 data register	P0DI	P0D	R	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	Port 0 mode register 01	P0MOD0	P0MOD01	R/W	8/16	0x05
0xF203		P0MOD1		R/W	8	0x00
0xF204	Port 0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x00
0xF205		P0MOD3		R/W	8	0x00
0xF206	Port 0 mode register 45	P0MOD4	P0MOD45	R/W	8/16	0x00
0xF207		P0MOD5		R/W	8	0x00
0xF208	Port 0 mode register 67	P0MOD6	P0MOD67	R/W	8/16	0x00
0xF209		P0MOD7		R/W	8	0x00
0xF20A	Reserved register	-	-	-	-	-
0xF20B						
0xF20C	Reserved register	-	-	-	-	-
0xF20D						
0xF20E	Reserved register	-	-	-	-	-
0xF20F						
0xF210	Port 1 data register	P1DI	P1D	R	8/16	0xFF
0xF211		P1DO		R/W	8	0x00
0xF212	Port 1 mode register 01	P1MOD0	P1MOD01	R/W	8/16	0x00
0xF213		P1MOD1		R/W	8	0x00
0xF214	Port 1 mode register 23	P1MOD2	P1MOD23	R/W	8/16	0x00
0xF215		P1MOD3		R/W	8	0x00
0xF216	Port 1 mode register 45	P1MOD4	P1MOD45	R/W	8/16	0x00
0xF217		P1MOD5		R/W	8	0x00
0xF218	Port 1 mode register 67	P1MOD6	P1MOD67	R/W	8/16	0x00
0xF219		P1MOD7		R/W	8	0x00
0xF21A	Port 1 pulse mode register	P1PMDL	P1PMD	R/W	8/16	0x00
0xF21B		P1PMDH		R/W	8	0x00
0xF21C	Port 1 pulse selection register	P1PSLL	P1PSL	R/W	8/16	0x00
0xF21D		P1PSLH		R/W	8	0x00
0xF21E	Reserved register	-	-	-	-	-
0xF21F						
0xF220	Port 2 data register	P2DI	P2D	R	8/16	0xFF
0xF221		P2DO		R/W	8	0x00
0xF222	Port 2 mode register 01	P2MOD0	P2MOD01	R/W	8/16	0x00
0xF223		P2MOD1		R/W	8	0x00
0xF224	Port 2 mode register 23	P2MOD2	P2MOD23	R/W	8/16	0x00
0xF225		P2MOD3		R/W	8	0x00
0xF226	Reserved register	-	-	-	-	-
0xF227						

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF228	Reserved register	-	-	-	-	-
0xF229						
0xF22A	Port 2 pulse mode register	P2PMDL	P2PMD	R/W	8/16	0x00
0xF22B		P2PMDH		R/W	8	0x00
0xF22C	Port 2 pulse selection register	P2PSLL	P2PSL	R/W	8/16	0x00
0xF22D		P2PSLH		R/W	8	0x00

Table 17-2 List of Registers / Bits
(●:available/-:unavailable)

Port Name	Pin Name	Control register / bit						Available/ Unavailable	
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	20 pin product	24 pin product
Port 0	P00	P00DO	P00DI	P0MOD0	-	-	-	●	●
	P01	P01DO	P01DI	P0MOD1	-	-	-	●	●
	P02	P02DO	P02DI	P0MOD2	-	-	-	●	●
	P03	P03DO	P03DI	P0MOD3	-	-	-	●	●
	P04	P04DO	P04DI	P0MOD4	-	-	-	●	●
	P05	P05DO	P05DI	P0MOD5	-	-	-	●	●
	P06	P06DO	P06DI	P0MOD6	-	-	-	●	●
	P07	P07DO	P07DI	P0MOD7	-	-	-	-	●
Port 1	P10	P10DO	P10DI	P1MOD0	-	-	-	●	●
	P11	P11DO	P11DI	P1MOD1	-	-	-	●	●
	P12	P12DO	P12DI	P1MOD2	-	-	-	●	●
	P13	P13DO	P13DI	P1MOD3	P13PLVL	P13PEN	P13PSL	●	●
	P14	P14DO	P14DI	P1MOD4	-	-	-	●	●
	P15	P15DO	P15DI	P1MOD5	-	-	-	●	●
	P16	P16DO	P16DI	P1MOD6	-	-	-	●	●
	P17	P17DO	P17DI	P1MOD7	-	-	-	●	●
Port 2	P20	P20DO	P20DI	P2MOD0	-	-	-	-	●
	P21	P21DO	P21DI	P2MOD1	-	-	-	-	●
	P22	P22DO	P22DI	P2MOD2	-	-	-	-	●
	P23	P23DO	P23DI	P2MOD3	P23PLVL	P23PEN	P23PSL	●	●

17.2.2 Port n Data Register (PnD:n=0 to 2)

PnD is a SFR to read the level of the port n pin and write output data.
The input level of the port n pins can be read by reading PnDI in the input mode. Data written to PnDO in the output mode are output to the port n pins. The data written to PnDO is readable. The bit can be set when output is enabled or disabled. Enable or disable the input or output by using the port n mode register. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnDO that have no corresponding pin.

Address: 0xF200(P0DI/P0D), 0xF201(P0DO), 0xF210(P1DI/P1D), 0xF211(P1DO), 0xF220(P2DI/P2D), 0xF221(P2DO)
Access: R/W
Access size: 8/16 bit
Initial value: 0x00FF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnD															
Byte	PnDO								PnDI							
Bit	Pn7DO*	Pn6DO*	Pn5DO*	Pn4DO*	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7DI*	Pn6DI*	Pn5DI*	Pn4DI*	Pn3DI	Pn2DI	Pn1DI	Pn0DI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

*: n=2 (port 2) is not available.

Bit No.	Bit symbol name	Description
15 to 8	Pn7DO to Pn0DO	Set the output level of port n pins. 0: Output "L" (Initial value) 1: Output "H" * P27DO to P24DO bits return 0 when read.
7 to 0	Pn7DI to Pn0DI	Set the input level of port n pins. 0: The input level is "L" 1: The input level is "H" (Initial value) * P27DO to P24DO bits return 0 when read.

17.2.3 Port 0 Mode Register 01 (P0MOD01)

P0MOD01 is a SFR to select the input/output mode, input/output status, and shared function of P00 pin and P01 pin.

Address: 0xF202(P0MOD0/P0MOD01), 0xF203(P0MOD1)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0005

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	P0MOD01															
Byte	P0MOD1								P0MOD0							
Bit	P01PD	P01MD2	P01MD1	P01MD0	P01OD	P01PU	P01OE	P01IE	-	-	-	-	-	P00PU	P00OE	P00IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit No.	Bit symbol name	Description
15	P01PD	Select whether the output type of the P01 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to P01OD (initial value) 1: P-channel open drain output (P01OD setting is disabled)
14 to 12	P01MD2 to P01MD0	Select the shared function of P01 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
11	P01OD	Select the output type of P01 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
10	P01PU	Select the internal pull-up resistor of P01 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of P01PD, P01OD, P01OE, and P01IE bits determine whether the contents of the P01PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
9	P01OE	Enable the output of P01 pins. 0: Disable the output (Initial value) 1: Enable the output
8	P01IE	Enable the input of P01 pins. 0: Disable the input (Initial value) 1: Enable the input
7 to 3	-	Reserved bits
2	P00PU	Enable the internal pull-up resistor of P00 pin. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	P00OE	Enable the output of P00 pin 0: Disable the output (Initial value) 1: Enable the output

Bit No.	Bit symbol name	Description
0	P00IE	Enable the input of P00 pin 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **P00 pin is initially configured as the input with pull-up resistor. If input "L" level at an initial setting, the input current flows.**
- **Be sure to set P0MOD01 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the P0MOD01 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the P01MD3-0 bits.**

17.2.4 Port n Mode Register 01 (PnMOD01: n=1, 2)

PnMOD01 is a SFR to select the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnMOD01 register that have no corresponding pins.

Address: 0xF212(P1MOD0/P1MOD01), 0xF213(P1MOD1),
0xF222(P2MOD0/P2MOD01), 0xF223(P2MOD1)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD01															
Byte	PnMOD1								PnMOD0							
Bit	Pn1PD	Pn1MD2	Pn1MD1	Pn1MD0	Pn1OD	Pn1PU	Pn1OE	Pn1IE	Pn0PD	Pn0MD2	Pn0MD1	Pn0MD0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Pn1PD	Select whether the output type of the Pn1 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn1OD (initial value) 1: P-channel open drain output (Pn1OD setting is disabled)
14 to 12	Pn1MD2 to Pn1MD0	Select the shared function of Pn1 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
11	Pn1OD	Select the output type of Pn1 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
10	Pn1PU	Enable the internal pull-up resistor of Pn1 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of the Pn1PD, Pn1OD, Pn1OE, and Pn1IE bits determine whether the contents of the Pn1PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
9	Pn1OE	Enable the output of Pn1 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn1IE	Enable the input of Pn1 pins. 0: Disable the input (Initial value) 1: Enable the input
7	Pn0PD	Select whether the output type of the Pn0 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn0OD (initial value) 1: P-channel open drain output (Pn0OD setting is disabled)

Bit No.	Bit symbol name	Description
6 to 4	Pn0MD2 to Pn0MD0	Select the shared function of Pn0 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
3	Pn0OD	Select the output type of Pn0 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn0PU	Select the internal pull-up resistor of Pn0 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of the Pn0PD, Pn0OD, Pn0OE, and Pn0IE bits determine whether the contents of the Pn0PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
1	Pn0OE	Select the output of Pn0 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn0IE	Select the input of Pn0 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMOD01 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD01 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the PnmMD3-0 bits.**

17.2.5 Port n Mode Register 23 (PnMOD23:n=0 to 2)

PnMOD23 is a SFR to select the input/output mode, input/output status, and shared function of Pn2 pin and Pn3 pin. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnMOD23 register that have no corresponding pins.

Address: 0xF204(P0MOD2/P0MOD23), 0xF205(P0MOD3), 0xF214(P1MOD2/P1MOD23), 0xF215(P1MOD3), 0xF224(P2MOD2/P2MOD23), 0xF225(P2MOD3)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD23															
Byte	PnMOD3								PnMOD2							
Bit	Pn3PD	Pn3MD2	Pn3MD1	Pn3MD0	Pn3OD	Pn3PU	Pn3OE	Pn3IE	Pn2PD	Pn2MD2	Pn2MD1	Pn2MD0	Pn2OD	Pn2PU	Pn2OE	Pn2IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Pn3PD	Select whether the output type of the Pn3 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn3OD (initial value) 1: P-channel open drain output (Pn3OD setting is disabled)
14 to 12	Pn3MD2 to Pn3MD0	Select the shared function of Pn3 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
11	Pn3OD	Select the output type of Pn3 pins. A LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
10	Pn3PU	Enable the internal pull-up resistor of Pn3 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of Pn3PD, Pn3OD, Pn3OE, and Pn3IE bits determine whether the contents of the Pn3PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
9	Pn3OE	Enable the output of Pn3 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn3IE	Enable the input of Pn3 pins. 0: Disable the input (Initial value) 1: Enable the input
7	Pn2PD	Select whether the output type of the Pn2 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn2OD (initial value) 1: P-channel open drain output (Pn2OD setting is disabled)

Bit No.	Bit symbol name	Description
6 to 4	Pn2MD2 to Pn2MD0	Select the shared function of Pn2 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
3	Pn2OD	Select the output type of Pn2 pins. A LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn2PU	Enable the internal pull-up resistor of Pn2 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of the Pn2PD, Pn2OD, Pn2OE, and Pn2IE bits determine whether the contents of the Pn2PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
1	Pn2OE	Enable the output of Pn2 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn2IE	Enable the input of Pn2 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set PnMOD23 registers before setting EICON0, EIMOD0 and IE1 registers. If setting PnMOD23 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on PnmMD3-0 bits.**

17.2.6 Port n Mode Register 45 (PnMOD45: n=0, 1)

PnMOD45 is a SFR to select the input/output mode, input/output status, and shared function of Pn4 pin and Pn5 pin. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnMOD45 register that have no corresponding pins.

Address: 0xF206(P0MOD4/P0MOD45), 0xF207(P0MOD5), 0xF216(P1MOD4/P1MOD45), 0xF217(P1MOD5)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD45															
Byte	PnMOD5								PnMOD4							
Bit	Pn5PD	Pn5MD2	Pn5MD1	Pn5MD0	Pn5OD	Pn5PU	Pn5OE	Pn5IE	Pn4PD	Pn4MD2	Pn4MD1	Pn4MD0	Pn4OD	Pn4PU	Pn4OE	Pn4IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Pn5PD	Select whether the output type of the Pn5 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn5OD (initial value) 1: P-channel open drain output (Pn5OD setting is disabled)
14 to 12	Pn5MD2 to Pn5MD0	Select the shared function of Pn5 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
11	Pn5OD	Select the output type of Pn5 pins. A LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
10	Pn5PU	Enable the internal pull-up resistor of Pn5 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of Pn5PD, Pn5OD, Pn5OE, and Pn5IE bits determine whether the contents of the Pn5PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
9	Pn5OE	Enable the output of Pn5 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn5IE	Enable the input of Pn5 pins. 0: Disable the input (Initial value) 1: Enable the input
7	Pn4PD	Select whether the output type of the Pn4 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn4OD (initial value) 1: P-channel open drain output (Pn4OD setting is disabled)

Bit No.	Bit symbol name	Description
6 to 4	Pn4MD2 to Pn4MD0	Select the shared function of Pn4 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
3	Pn4OD	Select the output type of Pn4 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn4PU	Enable the internal pull-up resistor of Pn4 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of Pn4PD, Pn4OD, Pn4OE, and Pn4IE bits determine whether the contents of the Pn4PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
1	Pn4OE	Enable the output of Pn4 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn4IE	Enable the input of Pn4 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set PnMOD45 registers before setting EICON0, EIMOD0 and IE1 registers. If setting PnMOD45 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on PnmMD3-0 bits.**

17.2.7 Port n Mode Register 67 (PnMOD67:n=0, 1)

PnMOD67 is a SFR to select the input/output mode, input/output status, and shared function of Pn6 pin and Pn7 pin. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnMOD67 register that have no corresponding pins.

Address: 0xF208(P0MOD6/P0MOD67), 0xF209(P0MOD7), 0xF218(P1MOD6/P1MOD67), 0xF219(P1MOD7)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD67															
Byte	PnMOD7								PnMOD6							
Bit	Pn7PD	Pn7MD2	Pn7MD1	Pn7MD0	Pn7OD	Pn7PU	Pn7OE	Pn7IE	Pn6PD	Pn6MD2	Pn6MD1	Pn6MD0	Pn6OD	Pn6PU	Pn6OE	Pn6IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Pn7PD	Select whether the output type of the Pn7 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn7OD (initial value) 1: P-channel open drain output (Pn7OD setting is disabled)
14 to 12	Pn7MD2 to Pn7MD0	Select the shared function of Pn7 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
11	Pn7OD	Select the output type of Pn7 pins. A LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
10	Pn7PU	Enable the internal pull-up resistor of Pn7 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of Pn7PD, Pn7OD, Pn7OE, and Pn7IE bits determine whether the contents of the Pn7PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Invalid (cannot be pulled up) X: 0/1 can be either
9	Pn7OE	Enable the output of Pn7 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn7IE	Enable the input of Pn7 pins. 0: Disable the input (Initial value) 1: Enable the input
7	Pn6PD	Select whether the output type of the Pn6 pin should be P-channel open drain. Please refer to the data sheet of each product for the output current characteristics. 0: Output type is set to Pn6OD (initial value) 1: P-channel open drain output (Pn6OD setting is disabled)

Bit No.	Bit symbol name	Description
6 to 4	Pn6MD2 to Pn6MD0	Select the shared function of Pn6 pins. For the details of the shared function, see Table 1-3 "Pin List" 000: Primary function (Initial value) 001: 2nd function 010: 3rd function 011: 4th function 100: 5th function 101: 6th function 110: 7th function 111: 8th function
3	Pn6OD	Select the output type of Pn6 pins. A LED is directly drive-able by enlarging the current when the N-channel open drain output mode is selected. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn6PU	Enable the internal pull-up resistor of Pn6 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The settings of Pn6PD, Pn6OD, Pn6OE, and Pn6IE bits determine whether the contents of the Pn6PU bits are reflected. 0X01: Valid (can be pulled up) 0111: Valid (can be pulled up) Other than the above: Not invalid (cannot be pulled up) X: 0/1 can be either
1	Pn6OE	Enable the output of Pn6 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn6IE	Enable the input of Pn6 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set PnMOD67 registers before setting EICON0, EIMOD0 and IE1 registers. If setting PnMOD67 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on PnmMD3-0 bits.**

17.2.8 Port n Pulse Mode Register (PnPMD:n=1, 2)

PnPMD is a SFR used when outputting a carrier frequency (pulse output) to the port n.
See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnPMD register that have no corresponding pin.

Address: 0xF21A(P1PMDL/P1PMD), 0xF21B(P1PMDH), 0xF22A(P2PMDL/P2PMD), 0xF22B(P2PMDH)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPMD															
Byte	PnPMDH								PnPMDL							
Bit	Pn7PLVL	Pn6PLVL	Pn5PLVL	Pn4PLVL	Pn3PLVL	Pn2PLVL	Pn1PLVL	Pn0PLVL	Pn7PEN	Pn6PEN	Pn5PEN	Pn4PEN	Pn3PEN	Pn2PEN	Pn1PEN	Pn0PEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	Pn7PLVL to Pn0PLVL	Select the condition of synchronizing level for outputting the carrier frequency to the pins. 0: Output the carrier frequency to the pins when the output level is "H" (initial value) 1: Output the carrier frequency to the pins when the output level is "L"
7 to 0	Pn7PEN to Pn0PEN	Enable or disable the pulse output of Pn7 to Pn0. These bits are valid when the Pn7 to Pn0 pins are configured as the output is enabled (Pn7OE to Pn0OE are "0"). 0: Disable the pulse output (initial value) 1: Enable the pulse output

17.2.9 Port n Pulse Selection Register (PnPSL:n=1, 2)

PnPSL is a SFR used to select the timer for generating the carrier frequency to the port n.
See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnPSL register that have no corresponding pin.

Address: 0xF21C(P1PSLL/P1PSL), 0xF21D(P1PSLH), 0xF22C(P2PSLL/P2PSL), 0xF22D(P2PSLH)
Accress: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPSL															
Byte	PnPSLH								PnPSLL							
Bit	-	-	-	-	-	-	-	-	Pn7PSL	Pn6PSL	Pn5PSL	Pn4PSL	Pn3PSL	Pn2PSL	Pn1PSL	Pn0PSL
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7 to 0	PnmPSL	Select the timer for generating the carrier frequency to the port nm. These bits are valid when the Pnm pins are configured as the output is enabled (PnmOE bits are "0"). n: port No.: 1,2 m: bit No.: 0 to 7 0: Operational timer 0 output (OTO0A) (Initial value) 1: Operational timer 1 output (OTO1A)

17.3 Description of Operation

The following shows description of port functions, where "n" is port number 0 to 2, and "m" is bit number 0 to 7.

17.3.1 Input

Each pin of port n sets PnmIE bit of PnMODm register to enter the state where input is enabled.
In the state with input enabled, the pin level can be read using PnDI. In addition, pull-up can be enabled by setting PnmPU bit of PnMODm register.
At a system reset, input disabled and no pull-up are selected as the initial status of pins except for the P00. As one of the P00, input, input enabled and pull-up are selected.

17.3.2 Output

Each pin of port n sets PnmOD bit of PnMODm register to select the output type of CMOS output or N-channel open drain or P-channel open drain and sets PnmOE bit of PnMODm register to enter the state where output is enabled.
In the state with output enabled, "L" or "H" level is output to each pin of the general-purpose port according to the value set in PnDO.
At a system reset, output disabled and CMOS output are selected as the initial status.
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.3 Primary Functions Other than Input/Output Function

External input (EXI0 to EXI3) can be used as the primary function other than the input/output function.
When using EXI0 to EXI3 as external interrupt input and the clock inputs of the 16-bit timer or trigger/clock input of the operational timer, set PnMODm register of the applicable port to input enabled (PnmIE bit="1").

See Chapter 18 "External Interrupt Control" for external interrupts, Chapter 8 "16-Bit Timer" for clock input of the 16-bit timer, and Chapter 9 "Operational Timer" for external trigger/clock input of the Operational timer.

17.3.4 Shared Function

Each pin of port n can use 2nd to 8th functions as the shared function.
Set PnmMD2 to PnmMD0 bits of PnMODm register to select each of the 2nd to 8th functions.

17.3.5 Carrier Frequency Output

17.3.5.1 Carrier Frequency Output Operation

A carrier frequency signal can be output from port n by setting PnPMD Register. See Table 17-2 "List of Registers/Bits" for pins supporting the carrier frequency output function. For the carrier frequency output, either of operational timer 0 output (OTO0A) or operational timer 1 output(OTO1A) can be used through setting PnPSL register.
See Chapter 9 "Operational Timer" for functional timer 0, 1 .
Figures 17-2 and 17-3 show an example of use of the carrier frequency output function.

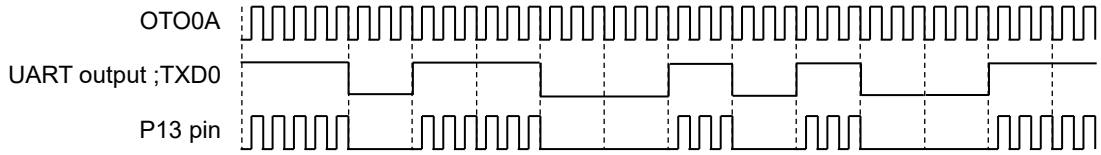


Figure 17-2 Example of Carrier Frequency Output When P13 Pin is Assigned to UART Output Pin
(P13PEN bit ="1", P13PLVL bit= "0" of P1PMD register)

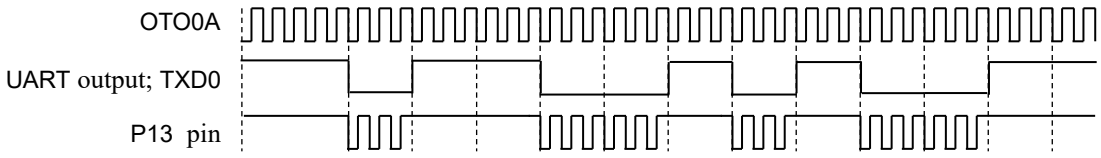


Figure 17-3 Example of Carrier Frequency Output When P13 Pin is Assigned to UART Output Pin
(P13PEN bit ="1", P13PLVL bit= "1" of P1PMD register)

17.3.5.2 Carrier Frequency Output Function Setting Procedure

Figure 17-4 shows an example of the carrier frequency output function setting procedure (with P13 pin used, TXD0 shared function, operational timer 0 output (OTO0A) used as a timer, carrier frequency output at "L" level).

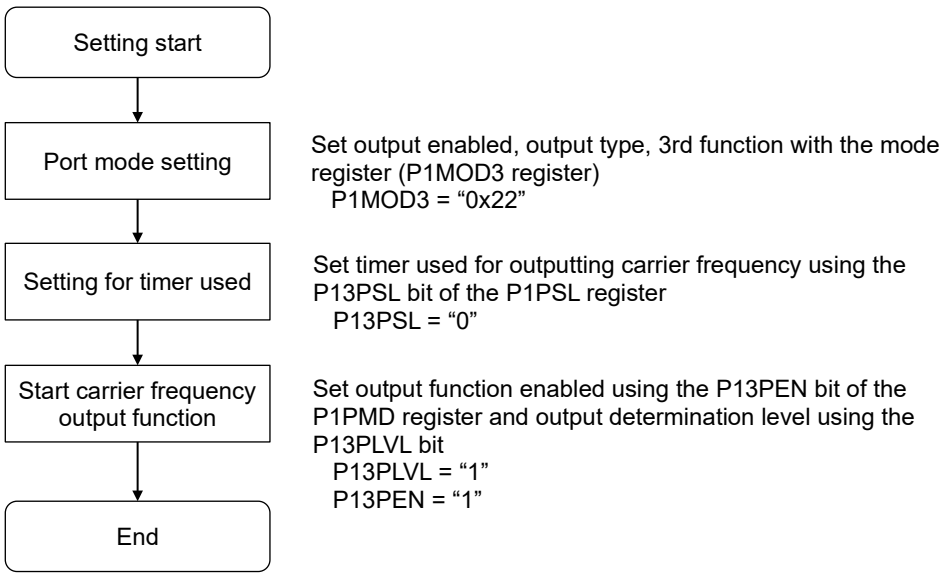


Figure 17-4 Example of Carrier Frequency Output Function Setting Procedure

17.3.6 Port Output Level Test

The level specified in PnDO can be read from PnDI by setting PnmOE bit of PnMODm register to "1" and PnmIE bit to "1". Use of this function allows confirmation that the level set in PnDO is being normally output to the port.

17.3.7 Port Setting Example

Figure 17-5 shows an example for setting port registers to output 0x55 to a port 1. It is also available to set output level before outputting.

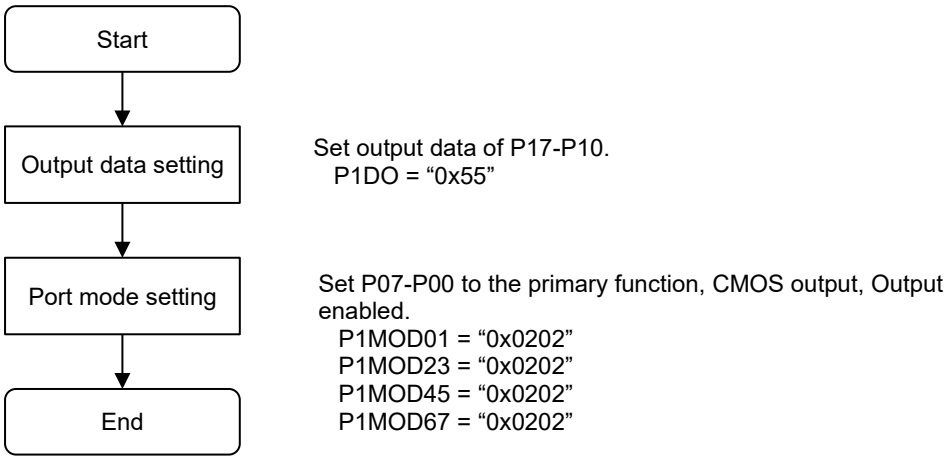


Figure 17-6 Setting example to output data to port 2

17.3.8 Notes for using P00/TEST0 pin

P00/TEST0 pin is used for the general port, the on-chip debug function or ISP function.
When using the on-chip debug function or ISP function, P00/TEST0 is unavailable to use as the general purpose port.
When using the general port, P00/TEST0 is unavailable to use for the on-chip debug function or ISP function.

Chapter 18 External Interrupt Function

18. External Interrupt Function

18.1 General Description

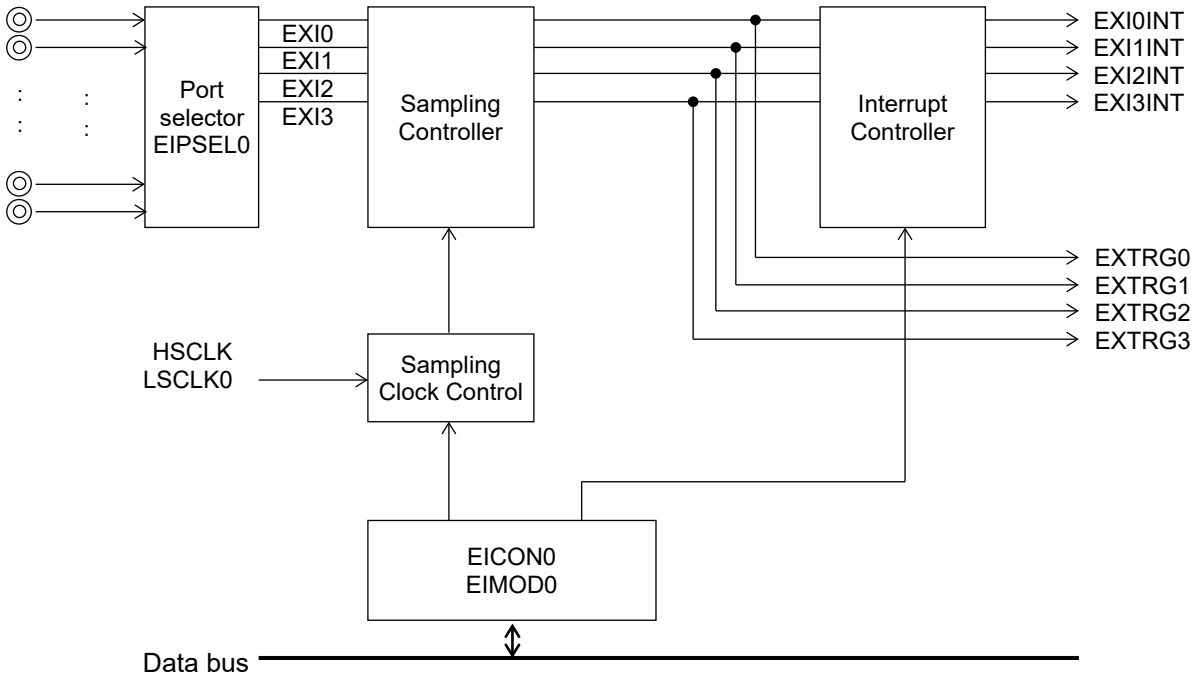
The external interrupt function generates interrupts by signals input to the general ports.
The interrupt channel has each dedicated interrupt vector. See Chapter 5 "Interrupt" for details of the interrupt vector.

18.1.1 Features

- Maskable 8 interrupts
- Each interrupt is assigned from max. 3 pins
- Selectable interrupt modes: interrupt disabled mode, falling-edge interrupt mode, rising-edge interrupt mode or both-edge interrupt mode
- Selectable "with sampling" or "without sampling" for the input signal (the sampling clock is LSCLK0 or HSCLK)

18.1.2 Configuration

Figure 18-1 shows the configuration of the external interrupt function (EXI0 to EXI3)



EICON0: External interrupt control register 0
EIMOD0: External interrupt mode register 0
EIPSEL0: External interrupt port select register 0

Figure 18-1 Configuration of External Interrupt Function

18.1.3 List of Pins

The external interrupt is assigned to the primary function of the general port.

Pin name	I/O	Function
EXI0	I	External interrupt input 0
EXI1	I	External interrupt input 1
EXI2	I	External interrupt input 2
EXI3	I	External interrupt input 3

Table 18-1 shows the list of the general ports used for the external interrupt and the register settings of the ports.

Table 18-1 Ports used for the external interrupt and the register settings

Pin name	Shared port		Setting register	Setting Value	Available	
					20 pin product	24 pin product
EXI0	P03	primary function	P0MOD3	0000_0X01*1	●	●
	P17	primary function	P1MOD7	0000_0X01*1	●	●
EXI1	P04	primary function	P0MOD4	0000_0X01*1	●	●
	P05	primary function	P0MOD5	0000_0001*1	●	●
EXI2	P06	primary function	P0MOD6	0000_0X01*1	●	●
	P13	primary function	P1MOD3	0000_0X01*1	●	●
EXI3	P00	primary function	P0MOD0	0000_0X01*1	●	●
	P10	primary function	P1MOD0	0000_0X01*1	●	●

●:Avilable -:Not avilable
*1: "X" determines the condition of the port input

X	Condition of port input
0	Input (without an internal pull-up resistor)
1	Input (with an internal pull-up resistor)

18.2 Description of Registers

18.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF044	External interrupt control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	-	-	-	-	-
0xF047	Reserved	-	-	-	-	-
0xF048	External interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A	Reserved	-	-	-	-	-
0xF04B	Reserved	-	-	-	-	-
0xF04C	External interrupt port selection register 0	EIPSEL0L	EIPSEL0	R/W	8/16	0x00
0xF04D		EIPSEL0H		R/W	8	0x00

18.2.2 External Interrupt Control Register 0 (EICON0)

EICON0 is a SFR to select the detection edge of the external interrupt input (EXI0 to EXI3).
Detecting the edge can generate the external interrupt (EXI0INT to EXI3INT).

Address : 0xF044(EICON0L/EICON0), 0xF045(EICON0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EICON0															
Byte	EICON0H								EICON0L							
Bit	-	-	-	-	PI3E1	PI2E1	PI1E1	PI0E1	-	-	-	-	PI3E0	PI2E0	PI1E0	PI0E0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	PI3E1 to PI0E1	Select the detection edge of the external interrupt (EXI0 to EXI7). 00 : Interrupt disabled (Initial value) 01 : Falling-edge interrupt 10 : Rising-edge interrupt 11 : Both-edge interrupt The relation of the bit number and the target external interrupt: Bit 11, 3 (PI3E1, PI3E0) : EXI3INT Interrupt Bit 10, 2 (PI2E1, PI2E0) : EXI2INT Interrupt Bit 9, 1 (PI1E1, PI1E0) : EXI1INT Interrupt Bit 8, 0 (PI0E1, PI0E0) : EXI0INT Interrupt
7 to 0	PI3E0 to PI0E0	

18.2.3 External Interrupt Mode Register 0 (EIMOD0)

EIMOD0 is a SFR to select the sampling clock and with/without sampling for the external interrupt (EXI0 to EXI3). Only one sampling clock can be chosen and it is shared for all the interrupt EXI0 to EXI3.

Address : 0xF048(EIMOD0L/EIMOD0), 0xF049(EIMOD0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EIMOD0															
Byte	EIMOD0H								EIMOD0L							
Bit	-	-	PG0DIV1	PG0DIV0	-	PG0CS0	-	-	-	-	-	-	PI3SM	PI2SM	PI1SM	PI0SM
R/W	R	R	R/W	R/W	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 14	-	Reserved bits
13 to 12	PG0DIV1 to PG0DIV0	Select frequency dividing ratio for the sampling clock in the EXI0 to EXI3. 00: No dividing (Initial value) 01: 1/2 of the sampling clock source 10: 1/4 of the sampling clock source 11: 1/8 of the sampling clock source
11	-	Reserved bits
10	PG0CS0	Select the sampling clock source in the EXI0 to EXI3. 0: LSCLK0 (Initial value) 1: HSCLK
9 to 4	-	Reserved bits
3 to 0	PI3SM to PI0SM	Select whether the input signals of EXI0 to EXI3 are detected with the sampling clock. 0: Detected without the sampling clock (Initial value) 1: Detected with the sampling clock except in STOP/STOP-D mode The relation of the bit number and the target external interrupt: Bit 3 (PI3SM) : EXI3INT Interrupt Bit 2 (PI2SM) : EXI2INT Interrupt Bit 1 (PI1SM) : EXI1INT Interrupt Bit 0 (PI0SM) : EXI0INT Interrupt

[Note]

- **If high-speed clock is selected as sampling clock source, it works without sampling when the high-speed clock does not supply; it includes stop by entry to standby mode. Set to LSCLK0 as sampling clock if needed.**
- **In the STOP mode, it works without sampling.**

18.2.4 External Interrupt Port Selection Register 0 (EIPSEL0)

EIPSEL0 is a SFR to select a port assigned to EXI0 to EXI3.

Address : 0xF04C(EIPSEL0L/EIPSEL0), 0xF04D(EIPSEL0H),
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EIPSEL0															
Byte	EIPSEL0H								EIPSEL0L							
Bit	-	-	-	-	-	-	-	-	-	EI3PS0	-	EI2PS0	-	EI1PS0	-	EI0PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15-0	EInPS0	Select a port assigned to EXI0 to EXI3. See Table 18-2 for detail. 0: Selection 0 (Initial value) 1: Selection 1

Table 18-2 assignment port to each EXI

EInPS0	EXI3	EXI2	EXI1	EXI0
0	P10	P06	P05	P03
1	P00	P13	P04	P17

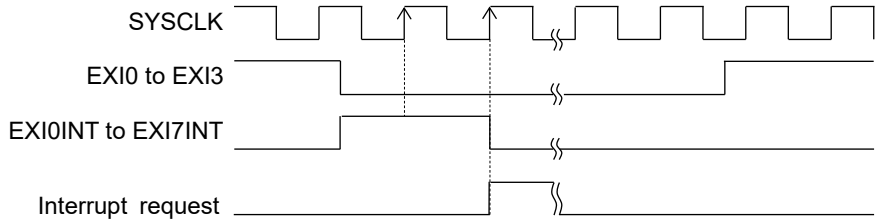
18.3 Description of Operation

18.3.1 Interrupt Request Timing

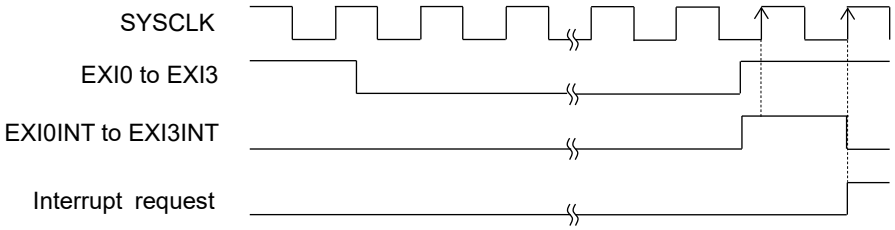
Figure 18-2 shows the interrupt generation timing without sampling (when the rising-edge/falling-edge/both-edge interrupt mode is chosen). Figure 18-3 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is chosen).
Table 18-3 shows the difference between the external interrupt generation timings with or without sampling after detection of the edge.

Table 18-3 EXI0INT to EXI11INT Generation After Detection of Edge of EXI0 to EXI3

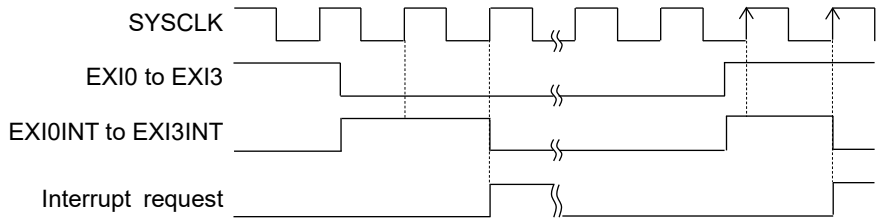
Sampling	Generation timing
No	Generated in synchronization with SYSCLK
Yes	Generated in synchronization with SYSCLK, when no transition for three periods with sampling clock after detecting edge.



(a) When falling-edge interrupt mode is chosen



(b) When rising-edge interrupt mode is chosen



(c) When both-edge interrupt mode is chosen

Figure 18-2 External Interrupt Generation Timing (without Sampling)

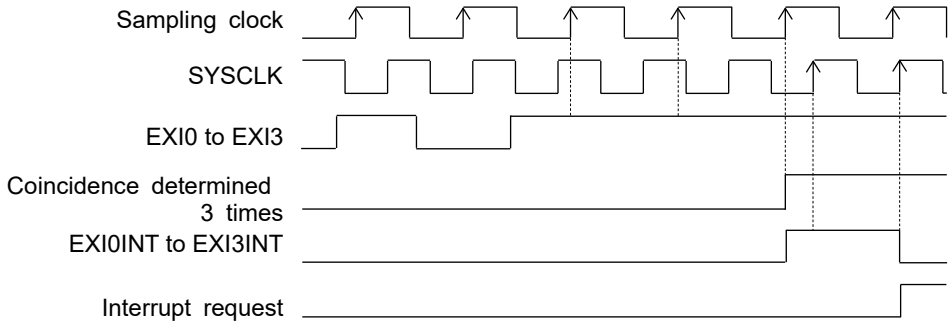
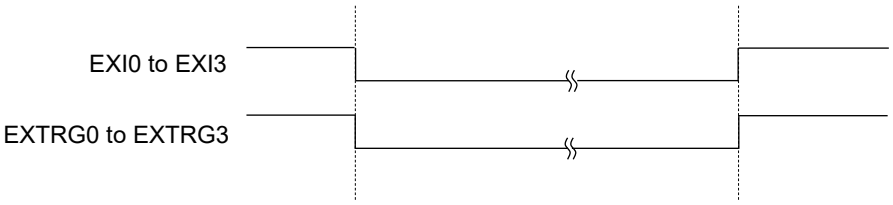


Figure 18-3 External Interrupt Generation Timing (with Sampling, with Rising-edge Interrupt Mode Chosen)

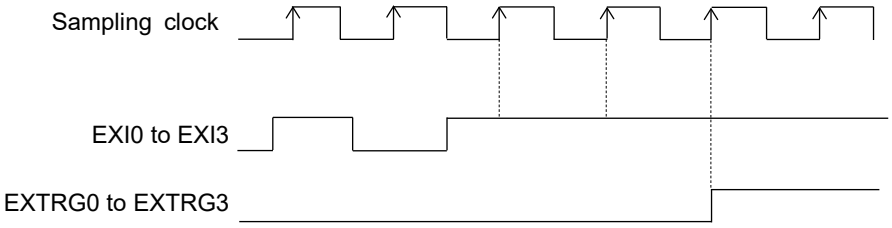
18.3.2 External Trigger Signal

Pins assigned with external interrupt can be used as external trigger signals (EXTRG0 to EXTRG3) for the 16-bit timer and function timer.
In addition, the sampling function contained in the external interrupt function can be used.

Figure 18-4 shows the external trigger signal timing.



(a) Without sampling



(b) With sampling

Figure 18-4 Operational Timer Trigger Signal

18.3.3 External Interrupt Setting Flow

Figure 18-5 shows the external interrupt setting flow.

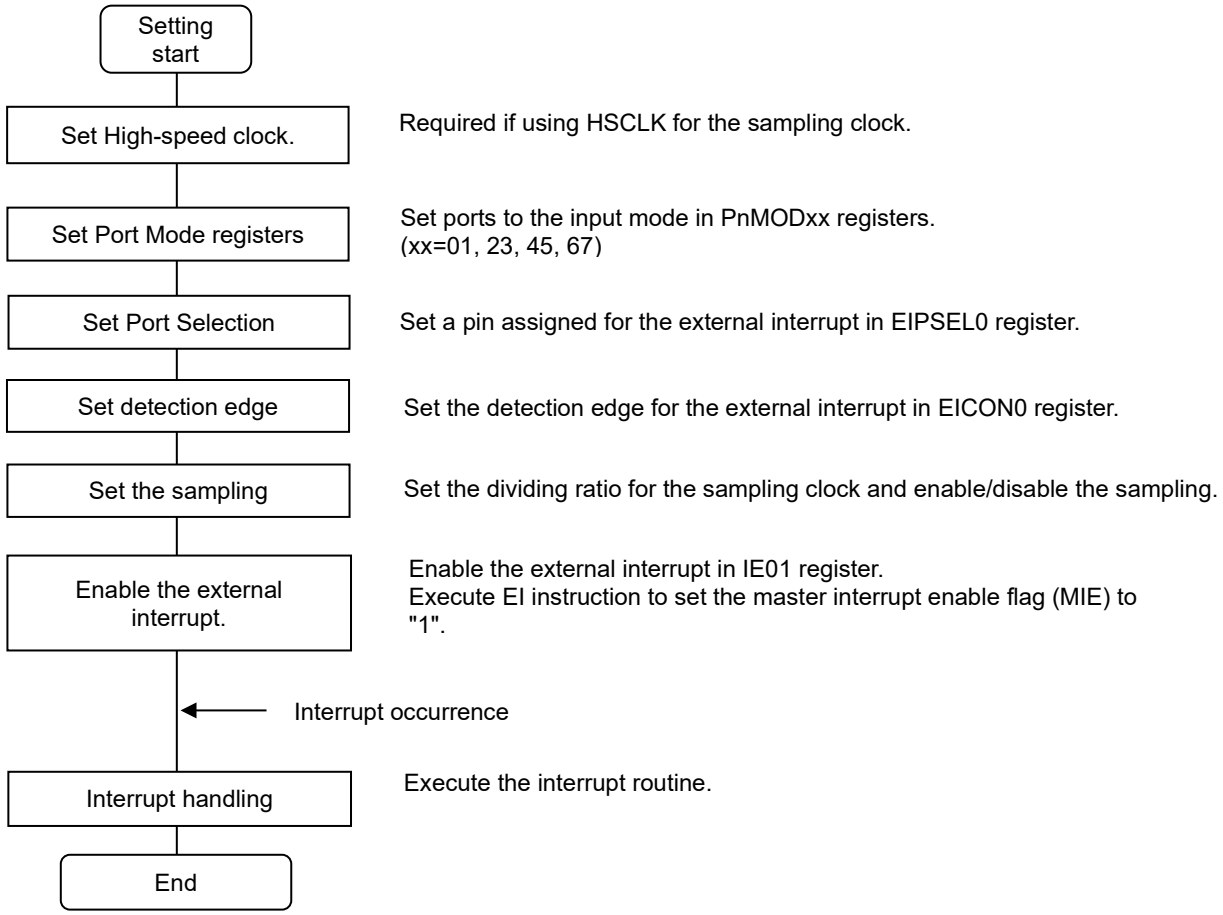


Figure 18-5 External Interrupt Setting Flow

Chapter 20 Analog Comparator

20. Analog Comparator

20.1 General Description

The Analog Comparator enables to use following functions.

- Compare voltages input to the two pins
- Compare a voltage input to the one pin with the internal reference voltage (CMP internal reference voltage (0.8V)/DAC0OUT/DAC1OUT).

Table 20-1 shows the number of channels.
Table 20-2 shows available internal reference voltage for each channel.

Table 20-1 Number of Analog Comparator channels

number of channels(n)	Available/Unavailable	
	20 pin product	24 pin product
0	●	●
1	●	●
2	●	●

●: Available -: Not available

Table 20-2 Available internally generated voltages for each channel

number of channels(n)	CMP internal reference voltage(0.8V)	DAC0OUT	DAC1OUT
0	-	●	●
1	●	●	●
2	●	-	●

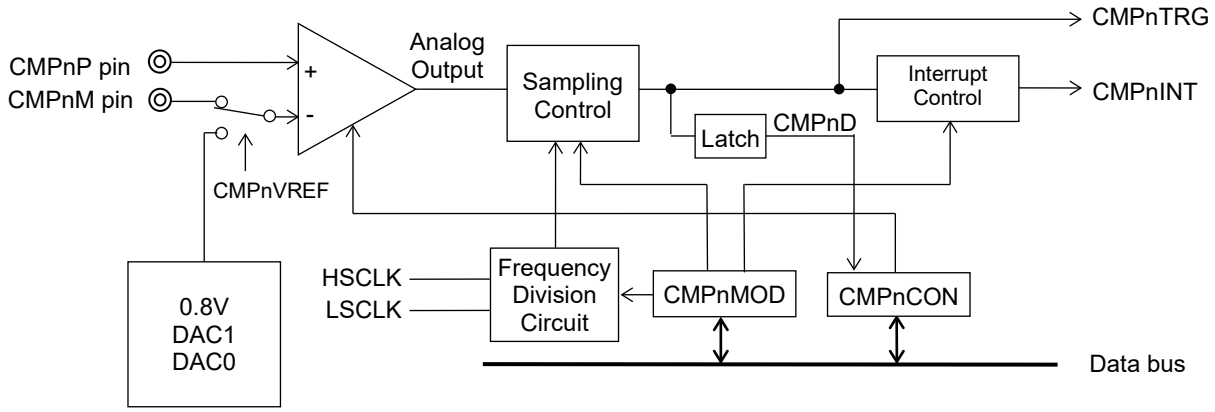
●: Available -: Not available

20.1.1 Features

- Comparable with external 2 voltage inputs.
- Comparable with external voltage input and internal reference voltage (CMP internal reference voltage (0.8V)/DAC0OUT/DAC1OUT).
- Three types of interrupt timing generated by the voltage comparison are available.
  - Rising edge of the comparison result
  - Falling edge of the comparison result
  - Rising edge and Falling edge of the comparison result
- The sampling with a clock is optional for the comparison result
  - HSCLK
  - LSCLK
  - 1/2 HSCLK to 1/ 8 HSCLK
  - 1/2 LSCLK to 1/ 8 HSCLK
- The analog comparator result output can be used as a trigger event source for operational timers and as a force stop trigger source.
- Same shared pin can be assigned to up to two different input channels of the analog comparator at the same time.
- Same shared pin can be assigned to the Analog comparator input and the AD converter input (AINn).
- Configurable the hysteresis Enable/disable setting of the analog comparator..

20.1.2 Configuration

Figure 20-1 shows the configuration of the analog comparator. (n=0 to 2)



CMPnCON : Comparator n control register
CMPnMOD : Comparator n mode register
CMPnD : Analog Comparator n result
CMPnINT : Analog Comparator n Interrupt
CMPnVREF : Reference voltage select setting
CMPnTRG : Analog Comparator n output

Figure 20-1 Configuration of Analog Comparator

20.1.3 List of Pins

The I/O pins of the Analog Comparator are assigned to the shared function of the general ports.

Pin name	I/O	Function
CMPnP	I	Analog comparator n non-inverting input
CMPnM	I	Analog comparator n inverting input

(n=0 to 2)

Table 20-3 shows the list of the general ports used for the Analog Comparator and the register settings of the ports.

Table 20-3 Ports used in the Analog Comparator and the register settings

Pin name				Shared port		Setting register	Setting Values	Available	
CMP0	CMP1	CMP2	ADC					20 pin product	24 pin product
CMP0P	-	-	--	P01	5th function/ 8th function	P0MOD1	0x40/ 0x70	●	●
	CMP1P	-	-		7th function		0x60	●	●
-	CMP1P *1	-	-	P14	5th function	P1MOD4	0x40	●	●
-		-	AIN0		8th function		0x70	●	●
-		CMP2P	-		7th function		0x60	●	●
-	-	CMP2P *3	-	P11	5th function/ 7th function/ 8th function	P1MOD1	0x40/ 0x60 0x70	●	●
CMP0M	-	-	-	P02	5th function/ 8th function	P0MOD2	0x40/ 0x70	●	●
	CMP1M	-	-		7th function		0x60	●	●
-	CMP1M *2	-	-	P15	5th function	P1MOD5	0x40	●	●
-		-	AIN1		8th function		0x70	●	●
-		CMP2M	-		7th function		0x60	●	●
-	-	CMP2M *4	-	P16	7th function	P1MOD6	0x40/ 0x60	●	●
-	-		AIN2		8th function		0x70	●	●

●: Available -: Not available

[Note]

- **When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers to set the general port to Hi-impedance, otherwise a shoot-through current may flow.**
- **An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.**
- ***1: If P01 is set to a 7th function, P14 cannot be used as CMP1P. (The setting will be disenabled)**
- ***2: If P02 is set to a 7th function, P15 cannot be used as CMP1M. (The setting will be disenabled)**
- ***3: If P14 is set to a 7th function, P11 cannot be used as CMP2P. (The setting will be disenabled)**
- ***4: If P15 is set to a 7th function, P16 cannot be used as CMP2M. (The setting will be disenabled)**

20.2 Description of Registers

20.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF880	Comparator control register	CMPCONL	CMPCON	R/W	8/16	0x00
0xF881		CMPCONH		R/W	8	0x00
0xF882	Comparator 0 mode register	CMP0MODL	CMP0MOD	R/W	8/16	0x00
0xF883		CMP0MODH		R/W	8	0x00
0xF884	Comparator 1 mode register	CMP1MODL	CMP1MOD	R/W	8/16	0x00
0xF885		CMP1MODH		R/W	8	0x00
0xF886	Comparator 2 mode register	CMP2MODL	CMP2MOD	R/W	8/16	0x00
0xF887		CMP2MODH		R/W	8	0x00
0xF888	Comparator status register	CMPSTA	-	R	8	0x00
0xF889	Reserved register	-		R	8	0x00

20.2.2 Comparator Control Register (CMPCON)

CMPnCON is a SFR to control the analog comparator.

Address: 0xF880(CMPCONL/CMPCON), 0xF881(CMPCONH)
Access: R/W
Access size: 8bit/16 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CMPCON															
Byte	CMPCONH								CMPCONL							
Bit	-	-	-	-	-	CMP2D	CMP1D	CMP0D	-	-	-	-	-	CMP2EN	CMP1EN	CMP0EN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial Varue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	CMP2D	Indicate the comparison result of the analog comparator 2. Indicate "0" when the analog comparator 2 is stopped. 0: CMP2P < CMP2M or internal reference voltage (initial value) 1: CMP2P > CMP2M or internal reference voltage
9	CMP1D	Indicate the comparison result of the analog comparator 1. Indicate "0" when the analog comparator 1 is stopped. 0: CMP1P < CMP1M or internal reference voltage (initial value) 1: CMP1P > CMP1M or internal reference voltage
8	CMP0D	Indicate the comparison result of the analog comparator 0. Indicate "0" when the analog comparator 0 is stopped. 0: CMP0P < CMP0M or internal reference voltage (initial value) 1: CMP0P > CMP0M or internal reference voltage
7 to 3	-	Reserved bits
2	CMP2EN	Control enable or disable the operation of the comparator 2. 0: Disables operating the comparator 2 (initial value) 1: Enables operating the comparator 2
1	CMP1EN	Control enable or disable the operation of the comparator 1. 0: Disables operating the comparator 1 (initial value) 1: Enables operating the comparator 1
0	CMP0EN	Control enable or disable the operation of the comparator 0. 0: Disables operating the comparator 0 (initial value) 1: Enables operating the comparator 0

[Note]

- **When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers ()n=0 to 2, m=0 to 7) to set the general port to Hi-impedance, otherwise a shoot-through current may flow.**
- **An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.**
- **After CMPnEN bit is set to "1", it takes 100µs to stabilize the operation of the analog comparator.**

20.2.3 Comparator n Mode Register (CMPnMOD: n=0,1,2)

CMPnMOD is a SFR to set the operation mode of the analog comparator.

Address: 0xF882(CMP0MODL/CMP0MOD), 0xF883(CMP0MODH),
0xF884(CMP1MODL/CMP1MOD), 0xF885(CMP1MODH),
0xF886(CMP2MODL/CMP2MOD), 0xF887(CMP2MODH),
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CMPnMOD															
Byte	CMPnMODH								CMPnMODL							
Bit	-	-	CMPnHYSEN	CMPnOMASK	CMPnOMASKV	-	CMPnVREF1	CMPnVREF0	-	-	CMPnDIV1	CMPnDIV0	CMPnCS1	CMPnCS0	CMPnE1	CMPnE0
R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Varue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15, 14	-	Reserved bits
13	CMPnHYSEN	Control enable or disable the operation of the hysteresis 0: Hysteresis disabled (initial value) 1: Hysteresis enabled
12	CMPnOMASK	When the analog comparator input is shared with the A/D converter input by setting 8th function as shown in Table1-3 Pin list, the output result of the analog comparator may fluctuate due to the influence of sampling in the AD conversion. By enabling this bit, the output of the analog comparator can be masked only while the A/D conversion. The output of the analog comparator is fixed to 0 or 1 by CMPnOMASKV bit when the mask setting is enabled. This masking function is enabled when 8th function is selected. It is disabled when other pin functions are selected. 0: Comparator output mask disabled (Initial value) 1: Comparator output mask enabled
11	CMPnOMASKV	Fixes the comparator output when CMPnOMASK bit enables the mask setting of the comparator output. 0: Fix the comparator output at "0" during the mask 1: Fix the comparator output at "1" during the mask
10	-	Reserved bit

Bit No.	Bit symbol name	Description
9, 8	CMPnVREF1, CMPnVREF0	Select the inverting input of the analog comparator.CH0 00: Voltage input from the CMPnM pin (initial value) 01: Do not use(CMPnM) 10: DAC0OUT 11: DAC1OUT CH1 00: Voltage input from the CMPnM pin (initial value) 01: CMP Internal reference voltage(0.8V) 10: DAC0OUT 11: DAC1OUT CH2 00: Voltage input from the CMPnM pin (initial value) 01: CMP Internal reference voltage (0.8V)10: Disable setting(DAC1OUT) 11: DAC1OUT
7, 6	-	Reserved bits
5, 4	CMPnDIV1, CMPnDIV0	Select frequency dividing ratio for the sampling clock in the analog comparator. 00: No dividing (Initial value) 01: 1/2 of the sampling clock source 10: 1/4 of the sampling clock source 11: 1/8 of the sampling clock source
3, 2	CMPnCS1, CMPnCS0	Select the sampling clock source in the analog comparator. 00: No sampling (Initial value) 01: Sampling with HSCLK 10: Sampling with LSCLK0 11: Do not use (No sampling)
1, 0	CMPnE1, CMPnE0	Select the timing of interrupt request generation. 00: Disable the interrupt request generation (initial value) 01: Falling-Edge 10: Rising-Edge 11: Both-Edge

[Note]

- **In the STOP mode, the sampling clock stops and the VLS works without sampling regardless the setting in CMPnCS1 bit and CMPnCS0 bit. When selecting "with sampling" and entering those mode, there is a time period(*1) in which interrupts gets disabled.**
  ***1 Time period to entering the STOP/STOP-D mode: Max.30µs. When returning from those modes, the interrupts are disabled until the sampling clock starts being supplied. The delay time depends on the configuration of clock and registers. See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".**
- **When the HSCLK is selected for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work. When using analog comparator in this case, select "No sampling" or "Sampling with LSCLK0" for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".**
- **Write CMPnMOD register when the Analog comparator stops (CMPnEN bit of CMPCON register is "0"), The analog comparator output is not guaranteed CMPnMOD register is changed while the analog comparator is activated.**
- **When the CMPnOMASK bit is set to "1", set to interrupt inhibition at CMPnE1/CMPnE0 bit. If interrupts are enabled, interrupts may occur unintentionally during mask processing at the start of A/D conversion.**
- **When the output of the D/A converter is selected as CMPnM pin and the DAC internal reference voltage (0.8V) is selected as the reference voltage of the D/A converter (DACMODE.DAMDn bit = "1"), enable sampling function of the analog comparator.**

20.2.4 Comparator Status Register(CMPSTA)

CMPSTA is a SFR to indicate the status of the analog comparator.

Adress: 0xF888(CMPSTA)
Access: R
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CMPSTA							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	CMP2 STA	CMP1 STA	CMP0 STA
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Varue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7~3	-	Reserved bits
2	CMP2STA	Indicate the operation history of analog comparator 2. After the analog comparator 2 is allowed to operate, the output of the analog comparator 2 changes to "1". It is cleared to "0" on the CMPSTA read. 0: No output change of analog comparator 2 (initial value) 1: Output change of analog comparator 2
1	CMP1STA	Indicate the operation history of analog comparator 1. After the analog comparator 1 is allowed to operate, the output of the analog comparator 1 changes to "1". It is cleared to "0" on the CMPSTA read. 0: No output change of analog comparator 1 (initial value) 1: Output change of analog comparator 1
0	CMP0STA	Indicate the operation history of analog comparator 0. After the analog comparator 0 is allowed to operate, the output of the analog comparator 0 changes to "1". It is cleared to "0" on the CMPSTA read. 0: No output change of analog comparator 0 (initial value) 1: Output change of analog comparator 0

[Note]

- **After the analog comparator is started, CMPnSTA bit may be set to "1" until the operation stabilizes. After the operation is stable, read CMPnSTA register once and initialize the value.**

20.3 Description of Operation

20.3.1 Analog Comparator Operation

Figure 20-2 shows an analog comparator operation overview.

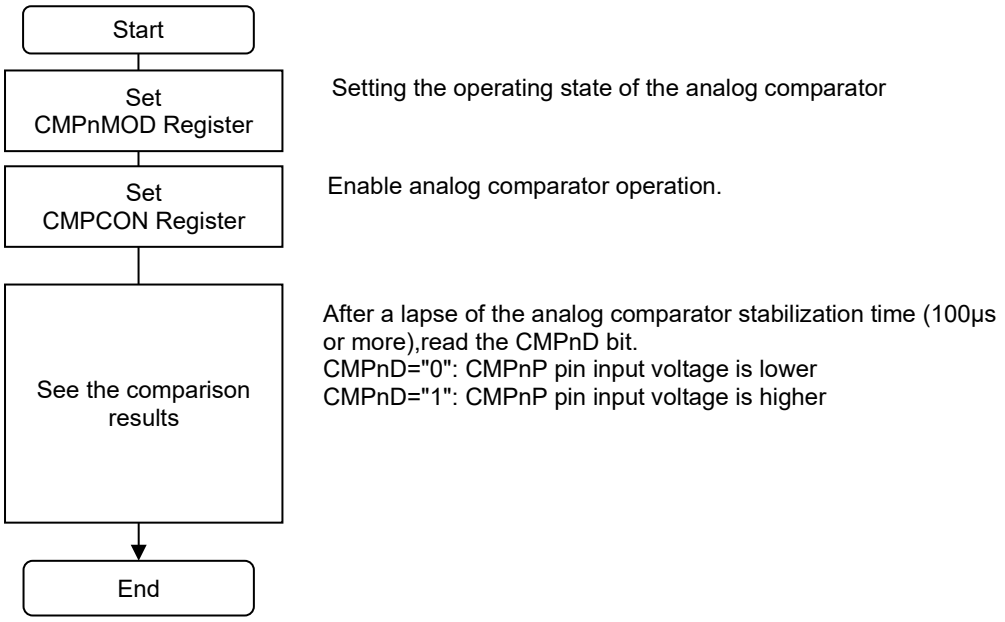


Figure 20-2 Analog Comparator Operation Overview

Figure 20-3 shows an example of the analog comparator operation timing.

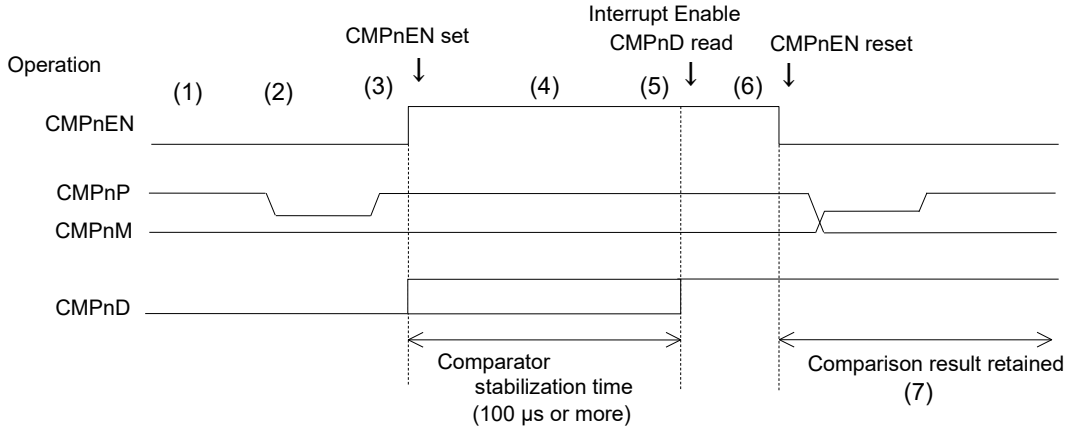


Figure 20-3 Example of Analog Comparator Operation Timing

Operation shown in Figure 20-3 above is described below.

(1) To operate the analog comparator, first set the following configuration:
   - Set the general-purpose port used for the analog comparator to high-impedance that both input and output is disabled, by writing "0" to PnmIE bit and PnmOE bit (m: bit number 0 to 7).
     Also, set the shared function of GPIO to the 5th, 7th, or 8th.
   - If using the HSCLK for the sampling clock, write "1" to the ENOSC bit of the frequency control register (FCON).

(2) Select the interrupt mode, sampling conditions, reference voltage, and hysteresis setting using the CMPnMOD register.

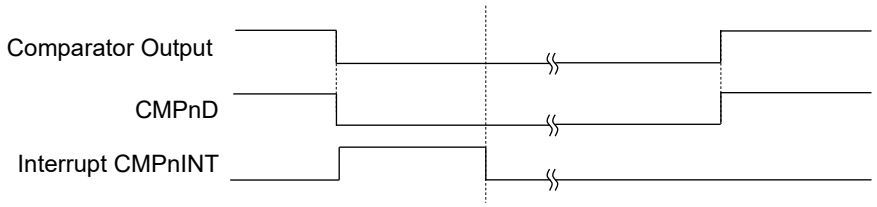
(3) Write "1" to the CMPnEN bit to enable the analog comparator operation.

(4) Wait for the stabilization time (100 µs or more) of the analog comparator.

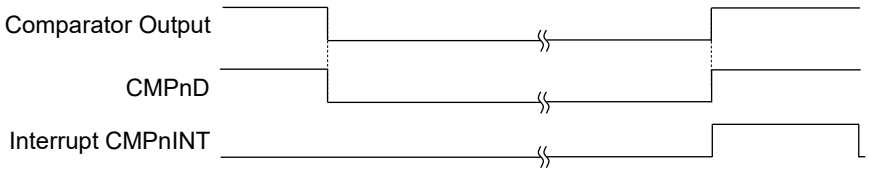
(5) Read the comparison result from CMPnD bit. If using interrupt, write "1" to the ECMPn bit of the interrupt enable register 45 (IE45), after clearing QCMPn bit of the interrupt request register 45 (IRQ45).
(6) Write "0" to CMPnEN bit to prohibit the operation of the analog comparator.

20.3.2 Interrupt Request

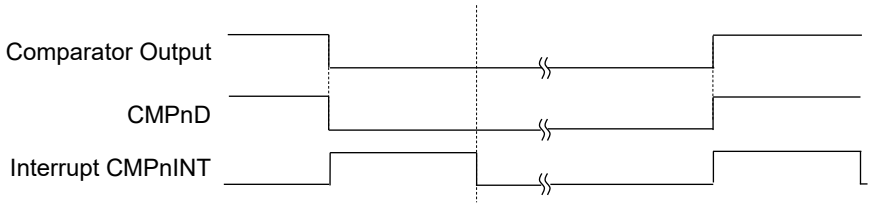
If the interrupt edge is selected in the CMPnE0 and CMPnE1 bits of the CMPnMOD register is detected, the analog comparator n interrupt (CMPnINT) is generated.
Figure 20-4 shows the interrupt generation timing without sampling (when the falling-edge/rising-edge/both-edge interrupt mode is selected).
Figure 20-5 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is selected).
Figure 20-6 shows the interrupt generation timing in the STOP mode.



(a) When falling-edge interrupt mode is selected



(b) When rising-edge interrupt mode is selected



(c) When both-edge interrupt mode is selected

Figure 20-4 Analog Comparator Interrupt Generation Timing (without Sampling)

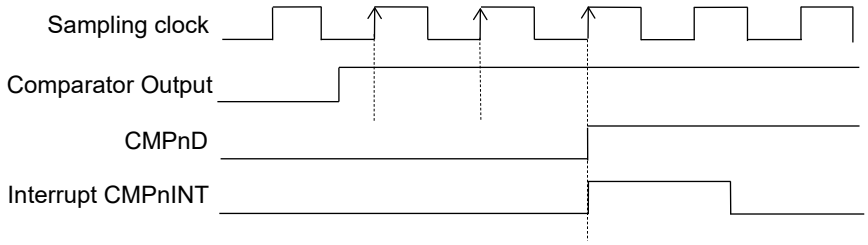


Figure 20-5 Analog Comparator Interrupt Generation Timing
(with Sampling, When Rising-edge Interrupt Mode is selected)

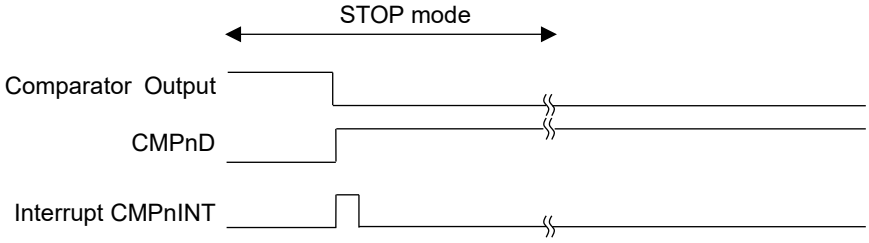


Figure 20-6 Analog Comparator Interrupt Generation Timing in STOP Mode
(When Falling-edge Interrupt Mode is selected)

20.3.3 Operation when sharing AINn pin and pin

By setting P14 to 8th function, the AIN0 pin and the CMP1P pin can be set to a shared pin.
By setting P15 to 8th function, the AIN1 pin and the CMP1M pin can be set to be a shared pin.
By setting P16 to 8th function, the AIN2 pin and the CMP2M pin can be set to a shared pin.
When the AINn and CMPnP/M pins are in the shared pin state, the analog comparator may respond unintentionally due to voltage fluctuations on the pins when A/D conversion is performed. To prevent unintended output of the analog comparator during A/D conversion, use the CMPnOMASK and CMPnMASKV bits of the CMPnMOD register to mask the output of the analog comparator during A/D conversion.

Figure 20-7-1 shows the operation waveform when AIN0 and CMP1P are used in the shared pin state, and the output of CMP1 is fixed to "H" during A/D conversion by the CMP1MOD register

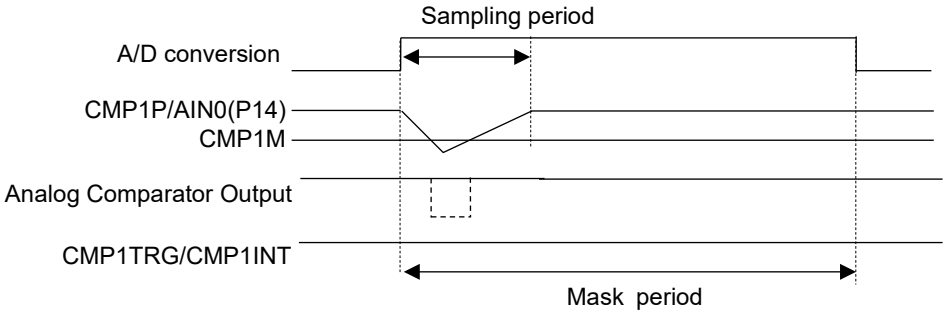


Figure 20-7-1 When sharing AIN0 and CMP1P
※Solid line is masked output waveform, dashed line is non-masked output waveform.

Figure 20-7-2 shows the operation waveforms when A/D conversion starts as soon as the analog comparator output goes “L”. As shown in the figure, if the analog comparator output before A/D conversion and the output after the analog comparator mask set by the CMPnMASKV bit in the CMPnMOD register are different, the comparator output may propagate as an extremely short pulse waveform. If a pulse waveform is generated, the analog comparator output may not be correctly transmitted inside the LSI.

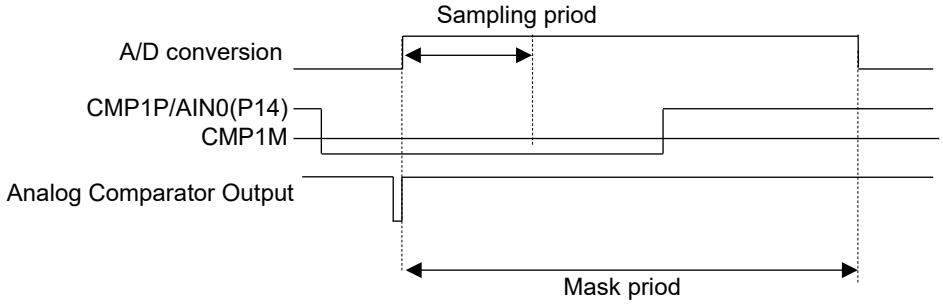
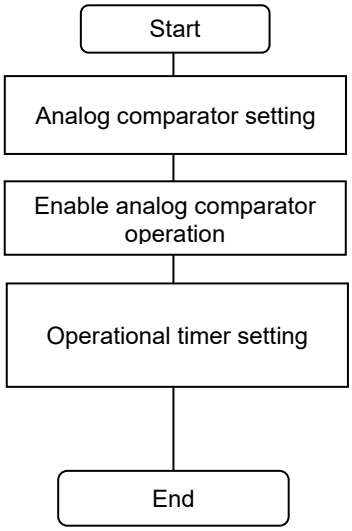


Figure 20-7-2 When sharing AIN0 and CMP1P

20.3.4 To output the analog comparator result from GPIO

Figure 20-8 shows the setting for outputting the output of the analog comparator from the pin.



CMPnMOD : Specify the analog comparator settings in the CMPnMOD register.
- ・CMPnHYSEN : Enable/disable hysteresis
- ・CMPnVREF1~0 : Setting comparator reference voltage

CMPCON : Select the analog comparator in the CMPCON register.

※After starting, please make waiting time for stable operation.

Set the operational timer n to the following
- ・OTnMOD : Set OTnMD0 bit to "0"
  Set OTnSTPOP/OTnSTPON bit to "1"
  Set all other bits to initial value
- ・OTnOCON : Select the analog comparator to output in the OTnOCON register.
- ・OTC0OCON/OTC1OCON : Set the OTnOCON setting to enabled

SFR other than the above is set to the initial value.

Figure 20-8 Setting to output analog comparator results from GPIO

Chapter 21 D/A Converter

21. D/A Converter

21.1 General Description

This LSI have one channel 8-bit resolution D/A converter that converts digital input signals to analog signals.
The number of D/A converter channels is dependent of the product specification.
Table 21-1 shows the number of channels.

Table 21-1 Number of D/A converter channels

Channel no.	Available	
	20pin product	24pin product
0	●	●
1	●	●

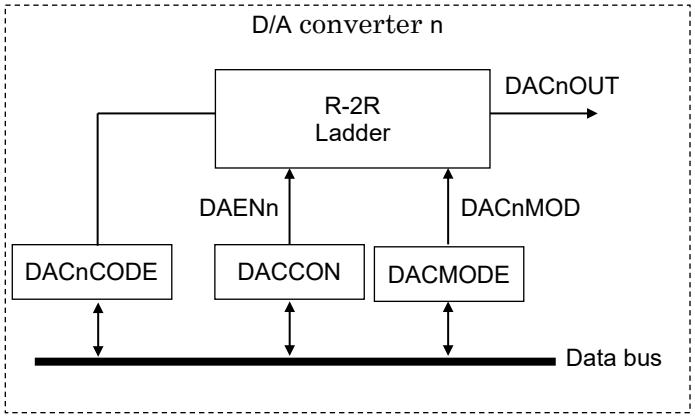
●: Available - : Not available

21.1.1 Features

- Number of channels : 2 channels
- 8-bit resolution
- R-2R ladder method
- Reference voltage (VDACREF0/VDACREF1) is selectable from V_{DD} or internal reference voltage (0.8V).
- Analog output (DAC0OUT)
  - Output voltage : $V_{DACREFn}$ ×(Setting value in SFR)/256 (n=0,1)
  - Output impedance : 10kΩ(Typ.)

21.1.2 Configuration

Figure 21-1 shows the configuration of the D/A converter.



DACCON : D/A converter control register
DACnCODE : D/A converter n code register
DACMODE : D/A converter mode register

∗DAC0OUT is connected to the output pin and the input of the analog comparator.
DAC1OUT is connected only to the input of the analog comparator.

Figure 21-1 Configuration of D/A Converter Circuit

21.1.3 List of Pins

The I/O pin (DAC0OUT) of the D/A converter 0 is assigned to the shared function of the general port.
The output pin (DAC1OUT) of the D/A converter 1 can only be used as an input of the analog comparator.

Pin name	I/O	Function
DAC0OUT	O	D/A converter 0 output

Table 21-2 shows the list of the general ports used for the D/A converter and the register settings of the ports.

Table 21-2 Ports used in the D/A converter and the register settings

Channel No.	Pin name	Shared port		Setting Register	Setting value	ML62Q2045 group	
						20pin product	24pin product
0	DAC0OUT	P23	5th function	P2MOD3	0100_0000	●	●

●: Available - : Not available

[Note]

- **For GPIO that outputs D/A converter, set the PnmMD2 to 0 bits (m: bit number 0 to 7) of the corresponding port n-mode register m to the 5th function, write "0" to the PnmIE bit and the PnmOE bit, and set it to high impedance (input disabled, output disabled). In other settings, the D/A converter is not output.**

21.2 Description of Registers

21.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF8A0	D/A converter control register	DACCON	-	R/W	8	0x00
0xF8A1	Reserved	-	-	-	-	-
0xF8A2	D/A converter mode register	DACMODE	-	R/W	8	0x00
0xF8A3	Reserved	-	-	-	-	-
0xF8A4	D/A converter 0 code register	DAC0CODE	-	R/W	8	0x00
0xF8A5	Reserved	-	-	-	-	-
0xF8A6	D/A converter 1 code register	DAC1CODE	-	R/W	8	0x00
0xF8A7	Reserved	-	-	-	-	-

21.2.2 D/A Converter Control Register (DACCON)

DACCON is a SFR to control the D/A converter.

Address: 0xF8A0 (DACCON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DACCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DAEN1	DAEN0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	DAEN1	Enable or disable the operation of the D/A converter 1. 0: Disable/Stop operating the D/A converter 1 (Initial value) 1: Enable operating the D/A converter 1
0	DAEN0	Enable or disable the operation of the D/A converter 0. 0: Disable/Stop operating the D/A converter 0 (Initial value) 1: Enable operating the D/A converter 0

[Note]

- **For GPIO that outputs D/A converter, set the PnmMD2 to 0 bits (m: bit number 0 to 7) of the corresponding port n-mode register m to the 5th function, write "0" to the PnmIE bit and the PnmOE bit, and set it to high impedance (input disabled, output disabled). In other settings, the converted result of D/A converter does not output.**

21.2.3 D/A Converter 0 Code Register (DACCODE) (n=0,1)

DACCODE is a SFR to set the conversion value of the D/A converter 0.

Address: 0xF8A4(DAC0CODE), 0xF8A6(DAC1CODE)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DACnCODE							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	d7 to d0	Set the output voltage of the D/A converter n. According to the reference voltage VDACREFn selected in the D/A converter mode register, output the voltage according to the calculation formula below. A voltage of $V_{DACREFn}$ x DACCODE / 256 is output. 0x00: Output V_{SS} (Initial value) 0x01 to 0xFF: Output a voltage = $V_{DACREFn}$ x (decimal code of DACCODE / 256)

21.2.4 D/A Converter Mode Register (DACMODE)

DACMODE is a SFR to set the reference voltage $V_{DACREF0}$/$V_{DACREF1}$ of the D/A converter.

Address: 0xF8A2(DACMODE)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DACMODE							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DAMD1	DAMD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
1	DAMD1	Set the reference voltage $V_{DACREF1}$ of D/A converter 1. 0: $V_{DACREF1}$=VDD pin (Initial value) 1: $V_{DACREF1}$=DAC Internal reference voltage (0.8V) ($V_{DACREFI}$)
0	DAMD0	Set the reference voltage $V_{DACREF0}$ of D/A converter 0. 0: $V_{DACREF0}$=VDD pin (Initial value) 1: $V_{DACREF0}$=DAC Internal reference voltage (0.8V) ($V_{DACREFI}$)

21.3 Description of Operation

21.3.1 D/A Converter Operation

Figure 21-2 shows the flow chart to control process of the D/A converter n.
Figure 21-3 shows the operation timing chart.

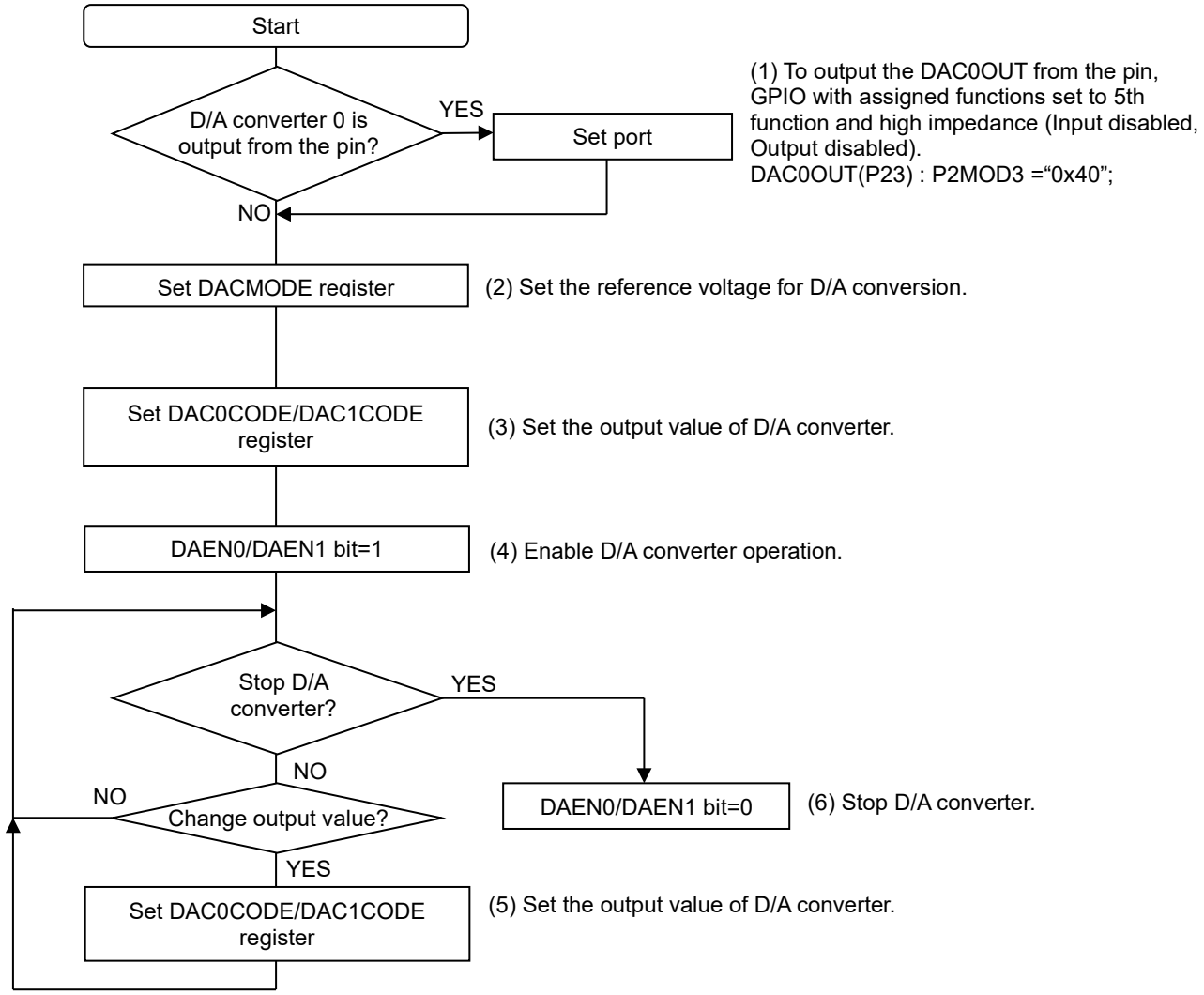
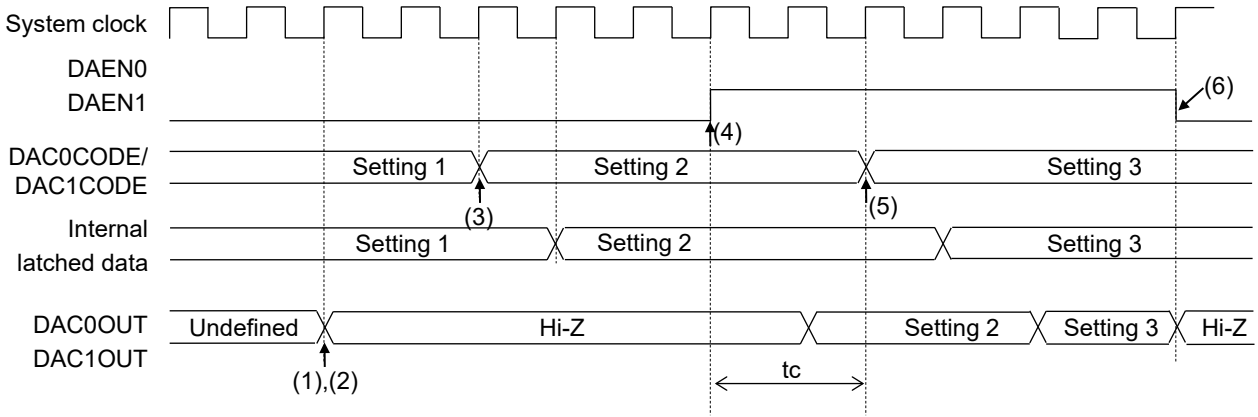


Figure 21-2 D/A Converter Process Flow Chart



tc : Conversion cycle of D/A converter

Figure 21-3 D/A Conversion Operation Timing Chart

Chapter 22 LOW Level Detector

22. Low Level Detector

22.1 General Description

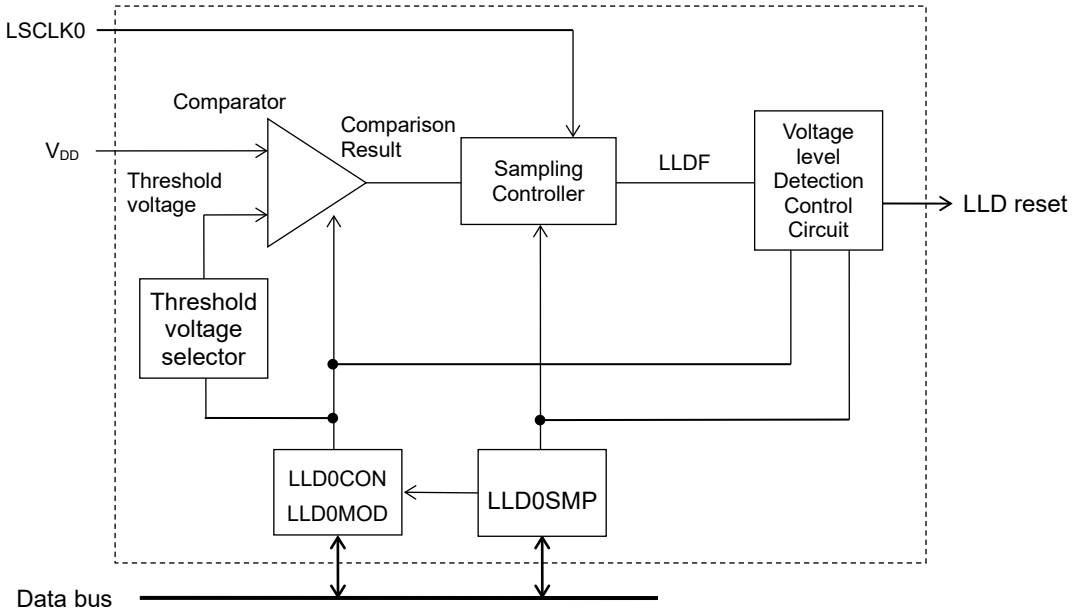
This LSI has Low Level Detector (LLD).
LLD generates a reset signal when the voltage level of V_{DD} is lower than threshold voltage.

22.1.1 Features

- Determine the voltage level of V_{DD} and cause reset when low voltage V_{DD} is detected
- Initialize by the power-on reset (POR) or pin reset

22.1.2 Configuration

Figure 22-1 shows the configuration of the LLD.



LLD0CON : Low Level Detector control register
LLD0MOD : Low Level Detector mode register
LLD0SMP : Low Level Detector sampling register

Figure 22-1 Configuration of Low Level Detector

22.2 Description of Registers

22.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF890	Low Level Detector control register	LLD0CON	-	R/W	8	0x00
0xF891	Reserved	-	-	-	-	-
0xF892	Low Level Detector mode register	LLD0MOD	-	R/W	8	0x00
0xF893	Reserved	-	-	-	-	-
0xF894	Reserved	-	-	-	-	-
0xF895	Reserved	-	-	-	-	-
0xF896	Low Level Detector sampling register	LLD0SMP	-	R/W	8	0x00
0xF897	Reserved	-	-	-	-	-

22.2.2 Low Level Detector control register (LLD0CON)

LLD0CON is a SFR to control the LLD.
This register is unresetable by anything other than the Power On Reset(POR) and RESETN pin reset.

Address: 0xF890 (LLD0CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LLD0CON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	LLD0RF	LLD0F	LLD0EN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2	LLD0RF	Indicate whether the LLD is in operation stably. 0: The LLD circuit is stopped or LLD is being stabilized (initial value) 1: LLD is in stable operation
1	LLD0F	Indicate whether the voltage level of V_{DD} is lower than threshold voltage. When the reset function is enabled by setting LLD0SEL0 bit of LLD0MOD register, a low level detector reset occurs when LLD0F bit becomes "1". This bit is cleared to "0" by writing "1" to this bit, but not cleared by writing "0". Also, this bit is cleared to "0" when the LLD starts operating. 0: The voltage level of V_{DD} is more than threshold voltage or the operation is stopped (initial value) 1: The power voltage (V_{DD}) is lower than the threshold voltage
0	LLD0EN	Control the LLD operation. When writing "1" to LLD0EN bit, write "1" to LLD0SEL0 bit of LLD0MOD register as well. 0: Disable operating the LLD (Initial value) 1: Enable operating the LLD

[Note]

- **When the LLD is in enable operating (LLD0EN=1) and shift to STOP, make sure that the LLD is in stable operation (LLD0RF bit = 1) before transitioning.**
- **If a reset other than Power On Reset and pin reset occurs during LLD operation, the LLD maintains its operating state.**

22.2.3 Low Level Detector mode register (LLD0MOD)

LLD0MOD is a SFR to controls the operation mode of the LLD.
Set this register only when the LLD is stopped (LLD0EN bit of LLD0CON register is "0").
This register is unresetable by anything other than the Power On Reset(POR) and RESETN pin reset.

Address: 0xF892 (LLD0MOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LLD0MOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rsvd	LLD0SEL0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	rsvd	Reserved bit Always write "0".
0	LLD0SEL0	Controls Disable / Enable LLD reset. For details on the conditions which an LLD reset occurs, see Section 22.3 "Description of Operation". 0: Disable reset function (Initial value) 1: Enable reset function

[Note]

- **When the LLD is in the enabled state (LLD0EN=1) and the STOP mode is shifted, make sure that the LLD is in stable operation (LLD0RF bit = 1) before shifting.**

22.2.4 Low Level Detector sampling register (LLD0SMP)

LLD0SMP is a SFR to execute the sampling of judgment result.
Set this register only when the LLD is stopped (LLD0EN bit of LLD0CON register is "0").
This register is unresetable by anything other than the Power On Reset(POR) and RESETN pin reset.

Address: 0xF896 (LLD0SMP)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LLD0SMP							
Bit	-	-	-	-	-	-	-	-	-	-	LLD0DIV1	LLD0DIV0	LLD0SM1	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit symbol name	Description
7 to 6	-	Reserved bits
5 to 4	LLD0DIV1 to LLD0DIV0	Select frequency dividing ratio for the sampling clock. 00: No dividing (Initial value) 01: 1/2 of the sampling clock source 00: 1/4 of the sampling clock source 01: 1/8 of the sampling clock source
3	LLD0SM1	Select the sampling clock source for detecting the voltage level. 0: No sampling (Initial value) 1: Sampling with LSCLK0
2 to 0	-	Reserved bits

[Note]
- **During STOP, this register is set "no sampling" regardless of LLD0SM1 bit setting.**

22.3 Description of Operation

LLD monitors the voltage level of V_{DD} and generates an LLD reset when V_{DD} falls below the threshold voltage.
Refer to the data sheet for the characteristics of the threshold voltage when the power supply voltage is falling or rising.

22.3.1 Operation setting procedure

During LLD operation, the voltage level of V_{DD} is constantly monitored and outputs a reset when a low voltage is detected.
Figure 22-2 shows the flow chart for starting the LLD.

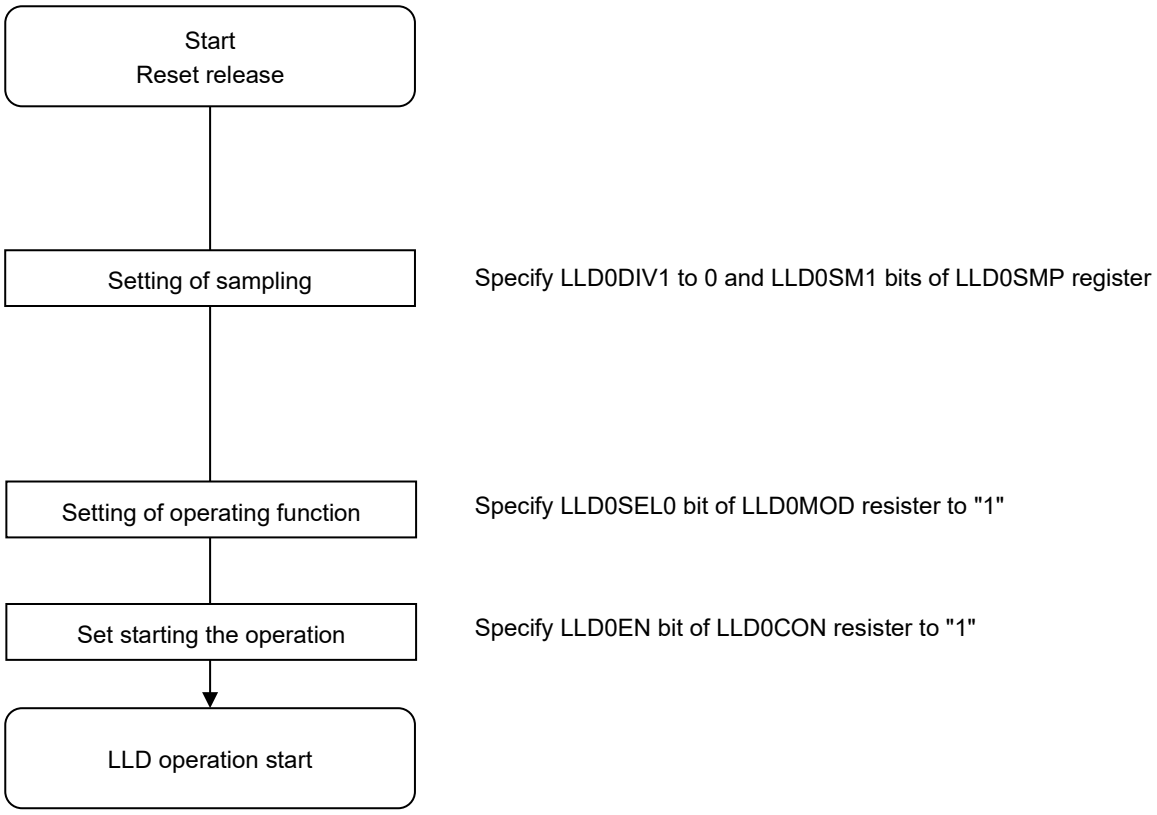


Figure 22-2 Flow chart for starting the LLD and enabling reset

22.3.1.1 Reset Output

Figure 22-3 shows the operating time chart when set to no sampling.

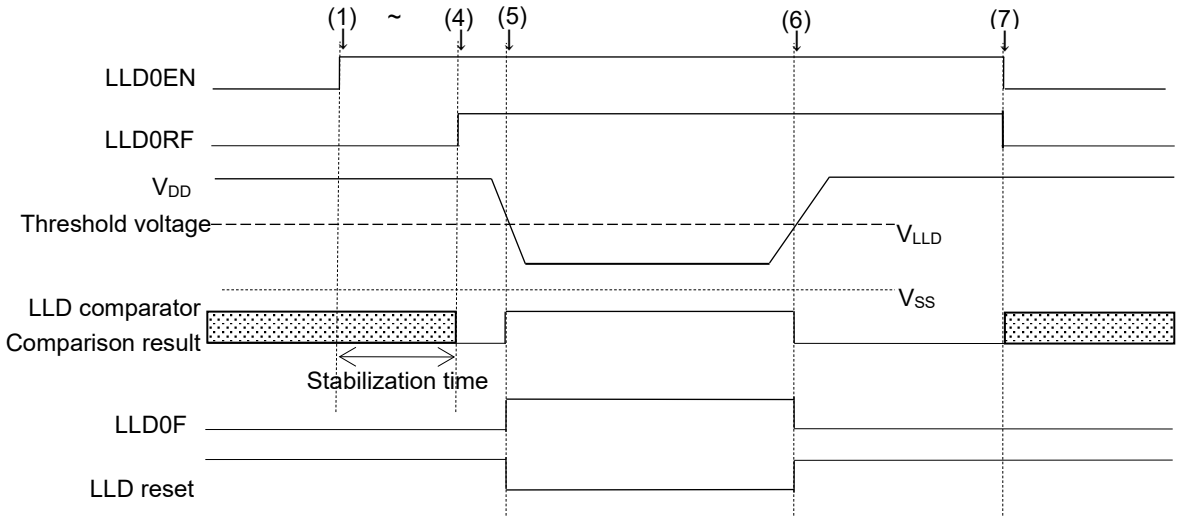


Figure 22-3 Operation Timing Chart When the LLD Reset Output without Sampling is specified

The operation shown in Figure 22-3 is described below:

(1) Choose "No sampling" by the LLD0SM1 bit of LLD0SMP register.
(2) Write “0x1” to LLD0SEL0 bit of the LLD0MOD register in order to enable the LLD reset.
(3) Set the LLD0EN bit of the LLD0CON register to "1".
(4) After approximately 300µs passed, the detection result of LLD becomes stabilized and the LLD0RF bit of the LLD0CON register is set to "1" (value of the voltage level supervisor bit (LLD0F) is read in software) (*1).
(5) When the power voltage (V_{DD}) becomes below the threshold voltage V_{LLD}, the LLD0F bit is set to "1" to generate the LLD reset.
(6) If V_{DD} becomes equal to or above the threshold voltage (V_{LLD}), the LLD0F bit is cleared to "0" to release the LLD reset.
(7) Write "0" to the LLD0EN bit to disable LLD operation.

*1 : The LLD0F bit/reset is masked until the LLD0RF bit becomes "1".

Figure 22-4 shows the operation timing chart when the LLD reset output with sampling is specified.

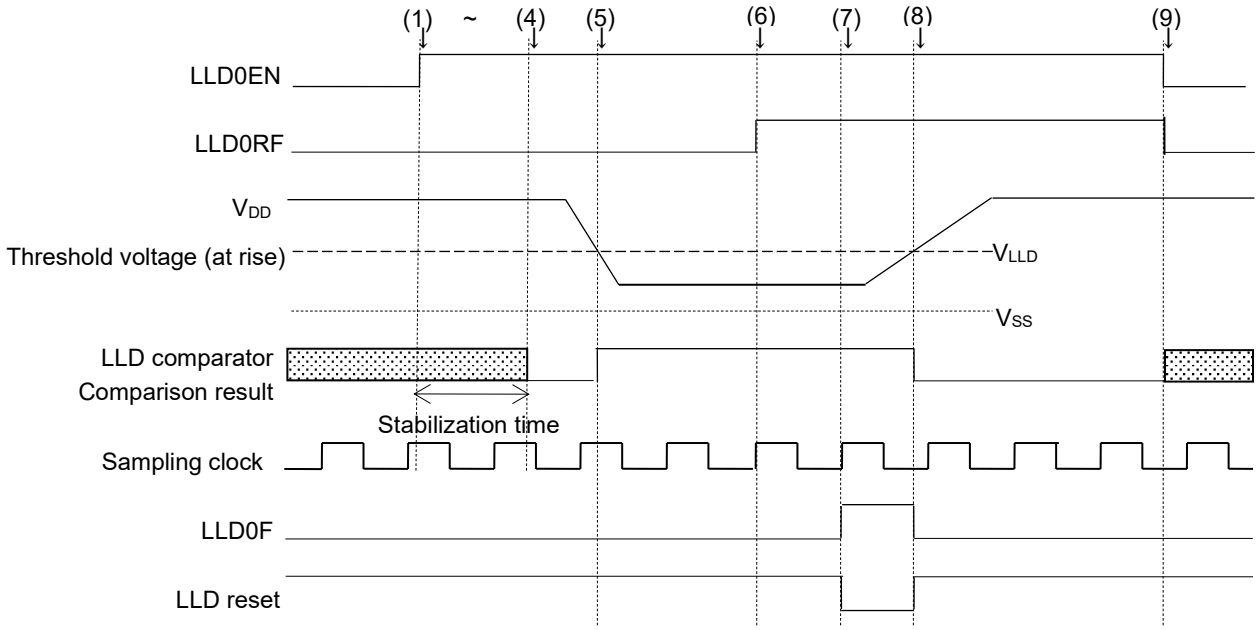


Figure 22-4 Operation Timing Chart When the LLD Reset Output with Sampling is specified

The operation shown in Figure 22-4 is described below:

(1) Select "Sampling with LLD0SM1" by the LLD0SM1 bit of LLD0SMP register. And specify sampling clock dividing ratio by the LLD0DIV1 to LLD0DIV0 bits of the LLD0SMP register.
(2) Write “0x1” to LLD0SEL0 bit of the LLD0MOD register in order to enable the LLD reset.
(3) Write "1" to the LLD0EN bit to enable LLD operation.
(4) Wait until the comparison result of the LLD comparator is stabilized (approx. 300µs).
(5) V_{DD} becomes below the threshold voltage (V_{LLD}).
(6) Once the comparison result of the LLD comparator is stabilized, the LLD0RF bit is set to "1" after three cycles of the sampling clock.
(7) If the comparison result of the LLD comparator is below the threshold voltage (V_{LLDF}) and this condition continues for the duration of three cycles or more of the sampling clock, the LLD0F bit is set to "1" and the LLD reset is generated.
(8) If the comparison result of the LDD comparator is equal to or above the threshold voltage (V_{LLD}) and this condition continues for the duration of three cycles or more of the sampling clock, the LLD0F bit is cleared to "0" and the LLD reset is canceled.
(9) Write "0" to the LLD0EN bit to disable LLD operation.

[Note]

- **It is impossible to change to STOP mode during the LLD stabilization time. When changing to STOP mode after LLD starting, make sure that the LLD0RF bit is set to "1" before changing STOP mode.**

Chapter 23 Successive Approximation Type A/D Converter

23. Successive Approximation Type A/D Converter

23.1 General Description

This LSI has the Successive Approximation type A/D Converter (SA-ADC), converts an analog input level to a digital value.
The number of A/D Converter channels is dependent of the product specification.
Table 23-1 shows the number of channels.

Table 23-1 Number of A/D Converter channels

Channel no.	Available	
	20pin	24pin
0	●	●
1	●	●
2	●	●
3	●	●
4	●	●

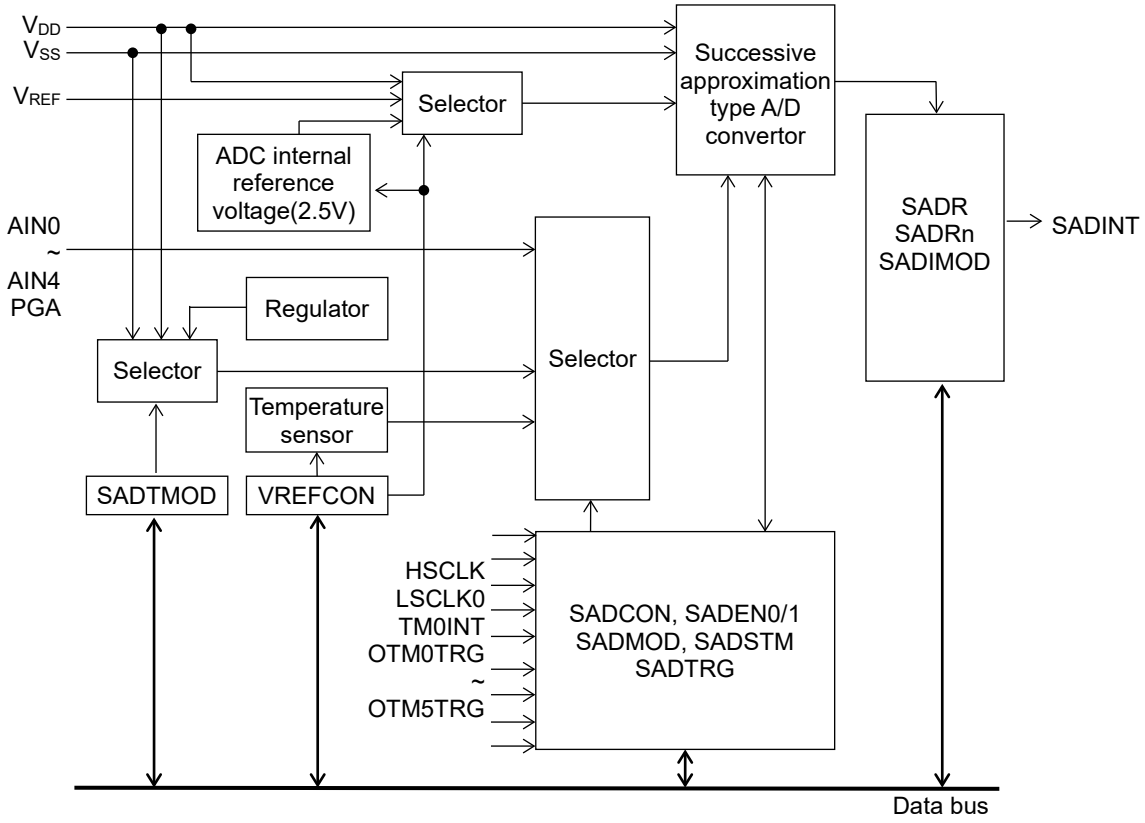
●: Available - : Not available

23.1.1 Features

- Resolution : 12bit
- Conversion time : Channel 0 to 4 : Min. 1.375µs/channel (conversion clock is 16MHz)
- Channel 16 (Temperature sensor) : Min. 10.5µs
- Channel 17(PGA) : Min. 8.000µs
- Number of input channel : Pin 5 channels + Internal 2 channels
- Available temperature conversion function
- AIN input voltage can be amplified in PGA and can be A/D conversion.
- Reference voltage : Voltage input from the VDD pin, Internal reference voltage (approx. 2.5V)
- Sampling time can be chosen
- Consecutive scan conversion function for target channels
- Consecutive scan conversion with a specific interval time
- One conversion result register for each channel
- A built-in temperature sensor usable for the low-speed RC oscillation adjustment
- A/D converter self test function (full scale, zero scale, A/D conversion function of ADC internal reference voltage(2.5V))
- Following triggers is available to start the A/D conversion
  - 16 bit timer 0 interrupt (TM0INT),
  - Operational timer n trigger (OTM0TRG~OTM5TRG)

23.1.2 Configuration

Figure 23-1 shows the configuration of SA-ADC.



SADCON	: SA-ADC control register
SADEN0/1	: SA-ADC enable register 0, 1
SADMOD	: SA-ADC mode register
SADSTM	: SA-ADC conversion interval register
SADR	: SA-ADC result register
SADRn	: SA-ADC result register n (n=0 to 4,16,17)
SADIMOD	: SA-ADC interrupt mode register
VREFCON	: Reference voltage control register
SADTRG	: SA-ADC trigger register
SADTMOD	: SA-ADC test mode register
SADINT	: SA-ADC interrupt request
TM0INT	: 16 bit timer 0 interrupt
OTM0TRG~OTM5TRG	: Operational timer 0 to 5 trigger

Figure 23-1 Configuration of successive approximation type A/D Converter

23.1.3 List of Pins

The I/O pins of the Successive Approximation type A/D converter are assigned to the shared function of the general ports.

Pin name	I/O	Description
VDD	-	Positive power supply for SA-ADC
VSS	-	Negative power supply for SA-ADC
AIN0	I	SA-ADC channel 0 analog input
AIN1	I	SA-ADC channel 1 analog input
AIN2	I	SA-ADC channel 2 analog input
AIN3	I	SA-ADC channel 3 analog input
AIN4	I	SA-ADC channel 4 analog input

Table 23-2 shows the list of the general ports used for the A/D Converter and the register settings of the ports.

Table 23-2 Ports used in the A/D Converter and the register settings

Channel no.	Pin name	Shared port		Setting Register	Setting value	Available/ unavailable	
						ML62Q2033 /2035	ML62Q2043 /2045
0	AIN0	P14	6/8th function	P1MOD4	0101_0000 /0111_0000	●	●
1	AIN1	P15	6/8th function	P1MOD5	0101_0000 /0111_0000	●	●
2	AIN2	P16	6/8th function	P1MOD6	0101_0000 /0111_0000	●	●
3	AIN3	P17	6/8th function	P1MOD7	0101_0000 /0111_0000	●	●
4	AIN4	P13	6/8th function	P1MOD3	0101_0000 /0111_0000	●	●
		P20	6/8th function	P2MOD0	0101_0000 /0111_0000	-	●

[Note]

- **When using the SA-ADC, set PnmMD2 to 0 bit to the 6th or 8th function of port n mode register 01/23/45/67 (n: port number 1, 2 m: bit number 0 to 7) and set PnmIE bit and PnmOE bit to "0" as "Disable input" and "Disable output", otherwise a shoot-through current may flow.**
- **While the A/D converter is operating, an influence of the noise is reducible by preventing the switching of neighboring pins or A/D converting in the HALT mode.**
- **When using AIN4, if both P13 and P20 are set to the 6th or 8th function, the input from P13 becomes AIN4 input. Input from P20 will be disabled.**

23.2 Description of Registers

23.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF800	SA-ADC mode register	SADMODL	SADMOD	R/W	8/16	0x30
0xF801		SADMODH		R/W	8	0x00
0xF802	SA-ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF803		SADCONH		R/W	8	0x00
0xF804	SA-ADC conversion interval register	SADSTML	SADSTM	R/W	8/16	0x00
0xF805		SADSTMH		R/W	8	0x00
0xF806	Reference voltage control register	VREFCON	-	R/W	8	0x00
0xF807	Reserved	-	-	-	-	-
0xF808	SA-ADC interrupt mode register	SADIMOD	-	R/W	8	0x00
0xF809	Reserved	-	-	-	-	-
0xF80A	SA-ADC trigger register	SADTRG	-	R/W	8	0x00
0xF80B	Reserved	-	-	-	-	-
0xF80C	SA-ADC enable register 0	SADEN0L	SADEN0	R/W	8/16	0x00
0xF80D		SADEN0H		R/W	8	0x00
0xF80E	SA-ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF80F		SADEN1H		R/W	8	0x00
0xF810 ~ 0xF82F	Reserved	-	-	-	-	-
0xF830	SA-ADC test mode register	SADTMOD	-	R/W	8	0x00
0xF831 ~ 0xF83D	Reserved	-	-	-	-	-
0xF83E	SA-ADC result register	SADRL	SADR	R	8/16	0x00
0xF83F		SADRH		R	8	0x00
0xF840	SA-ADC result register 0	SADR0L	SADR0	R	8/16	0x00
0xF841		SADR0H		R	8	0x00
0xF842	SA-ADC result register 1	SADR1L	SADR1	R	8/16	0x00
0xF843		SADR1H		R	8	0x00
0xF844	SA-ADC result register 2	SADR2L	SADR2	R	8/16	0x00
0xF845		SADR2H		R	8	0x00
0xF846	SA-ADC result register 3	SADR3L	SADR3	R	8/16	0x00
0xF847		SADR3H		R	8	0x00
0xF848	SA-ADC result register 4	SADR4L	SADR4	R	8/16	0x00
0xF849		SADR4H		R	8	0x00
0xF84A ~ 0xF85F	Reserved	-	-	-	-	-
0xF860	SA-ADC result register 16	SADR16L	SADR16	R	8/16	0x00
0xF861		SADR16H		R	8	0x00
0xF862	SA-ADC result register 17	SADR17L	SADR17	R	8/16	0x00
0xF863		SADR17H		R	8	0x00

23.2.2 SA-ADC Mode Register (SADMOD)

SADMOD is a SFR to set an operation mode and operating clock frequency for SA-ADC.

Address : 0xF800 (SADMODL/SADMOD), 0xF801 (SADMODH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0030

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADMOD															
Byte	SADMODH								SADMODL							
Bit	SAINITT3	SAINITT2	SAINITT1	SAINITT0	SAINIT	-	-	-	SASHT3	SASHT2	SASHT1	SASHT0	SACK2	SACK1	SACK0	SALP
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 12	SAINITT3 to SAINITT0	Configures amplifier stability time at conversion starting. wait time for stability [s] = setting value / SAD_CLK frequency This time should be equal or more than 0.5[µs] When SAINIT="1", it is included discharge time for sample hold capacitor. In this case, this time should be equal or more than 0.65[µs]. Make decision on the value with external impedance of input pin. Table 23-3 shows example for typical setting.
11	SAINIT	Control whether or not to discharge the electrical charge remained in the sample hold capacitor on the previous A/D conversion, before starting the next SA-ADC conversion. 0: Without discharging (Initial value) 1: With discharging
10 to 8	-	Reserved bits
7 to 4	SASHT3 to SASHT0	Set the sampling time. sampling time [s] = (setting value + 1) / SAD_CLK frequency Set the sampling time to more than 0.5[µs], and the setting value to more than 0x3. When A/D Conversion of PGA Input and temperature conversion are performed if the sampling time is set to less than 7[µs], the sampling time is fixed to 7[µs]. Table 23-4 shows example for typical setting.
3 to 1	SACK2 to SACK0	Select the frequency of the A/D conversion operating clock (SAD_CLK). If 1/1 x HSCLK is set in the SAD_CLK, it will automatically switch to 1/2 x HSCLK only when temperature conversion is in progress. 000: 1/1 x HSCLK (Initial value) 001: 1/2 x HSCLK 010: 1/4 x HSCLK 011: 1/8 x HSCLK 100: 1/16 x HSCLK 101: Do not use 110: Do not use 111: 1/1 x LSCLK0 The following formula is calculated A/D conversion time without discharge/amp. stability time. A/D conversion time [s] = (SASHT4 to SASHT0 value + 15) / SAD_CLK frequency
0	SALP	Select whether the A/D conversion is performed once only for each channel or consecutively. The conversion interval time in the consecutive scan A/D conversion mode is specified in the SADSTM register. 0: Single A/D conversion (Initial value) 1: Consecutive scan A/D conversion

Table 23-3 Example for SAINITT3 to 0 setting

SAD_CLK	SAINITT3 to 0 SAINIT=1 (Discharge time/Amp. stability) > 0.65µs	SAINITT3 to 0 SAINIT=0 (Amp. stability) > 0.5µs
16 MHz	1010	1000
8 MHz	0101	0100
4 MHz	0011	0010
< 4 MHz	0010	0010

Table 23-4 Example for SASHT3 to 0 setting

	SASHT3 to 0 (Sampling time)	
SAD_CLK	ADC internal reference voltage(2.5V)	V_{DD} reference
16 MHz	-*1	0111
Up to 8 MHz	0011	0011

*1: When the ADC internal reference voltage (2.5V) is selected as the reference voltage to be used for ADC conversion (VREFP0 bit in the VREFCON register = 1),16 MHz is not allowed to be set to SAD_CLK. If set, the conversion result cannot be guaranteed.

23.2.3 SA-ADC Control Register (SADCON)

SADCON is a SFR to control the operation of the A/D converter.

Address : 0xF802 (SADCONL/SADCON), 0xF803 (SADCONH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADCON															
Byte	SADCONH								SADCONL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SATGEN	SARUN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1	SATGEN	Enable starting the A/D conversion by the trigger events. 0: Disable the trigger operation (Initial value) 1: Enable the trigger operation
0	SARUN	Start or stop the A/D conversion. Write "1" to this bit to start the A/D conversion, and "0" to stop it. When "0" is written to SALP bit and the A/D conversion on the largest number of channel is ended, this SARUN bit is automatically reset to "0". When "1" is written to SALP, the A/D conversion repeats until the SARUN bit is reset to "0" by the software. 0: Stop the A/D conversion (Initial value) 1: Start the A/D conversion

[Note]

- **Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN0 and SADEN1). If no channel is chosen, the operation does not start.**
- **Enter STOP mode after checking SARUN bit is "0". It does not enter the STOP mode when SARUN bit is "1".**
- **When SACK2 to 0 bits are set to 0x7, it takes max. 3 clocks of the low-speed clock (LSCLK0) to start or stop the A/D conversion after setting or resetting SARUN bit.**

23.2.4 SA-ADC Conversion Interval Register (SADSTM)

SADSTM is a SFR to set the interval time in the consecutive scan A/D conversion mode.

Address : 0xF804 (SADSTML/SADSTM), 0xF805 (SADSTMH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADSTM															
Byte	SADSTMH								SADSTML							
Bit	SADSTM15	SADSTM14	SADSTM13	SADSTM12	SADSTM11	SADSTM10	SADSTM9	SADSTM8	SADSTM7	SADSTM6	SADSTM5	SADSTM4	SADSTM3	SADSTM2	SADSTM1	SADSTM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interval time is determined by the following formula.

A/D conversion interval time = HSCLK cycle x SADSTM setting value

For an example, supposing to A/D convert channel 2 and channel 5, the A/D conversion interval time means the time after the channel 2 and channel 5 are A/D converted consecutively and before the A/D conversion of channel 2 is started. The next A/D conversion starts at the timing that the value set in this register has been counted with SAD_CLK.

[Note]

- **If only temperature conversion is enabled and the SAD_CLK is set to 1/1 x HSCLK, the SAD_CLK period used for the conversion interval for A/D conversion will be 1/2 x HSCLK.**

23.2.5 Reference Voltage Control Register (VREFCON)

VREFCON is a SFR to select reference voltage for the SA-ADC.

Address : 0xF806 (VREFCON)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VREFCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	VREFP0	-	-	-	VREFEN
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4	VREFP0	Select the reference voltage for the A/D conversion. 0: V_{dd} pin (Initial value) 1: V_{REF} pin
3 to 1	-	Reserved bits
0	VREFEN	Enable the operation of the ADC internal reference voltage (2.5 V) and temperature sensor. When using the ADC internal reference voltage (2.5 V) and using the temperature sensor circuit, set the VREFEN bit to "1". 0: Disable the operation of internal reference voltage and temperature sensor (Initial value) 1: Enable the operation of internal reverence voltage and temperature sensor

[Note]

- **After VREFEN bit is set to "1", it takes 200µs (Max.) for the ADC internal reference voltage (2.5 V) to stabilize. A/D conversion should be started after this stabilization time has elapsed.**
  VREFEN bit is set to "1" and remains stable after 200µs until it is disabled. Therefore, there is no need to wait for 200µs (Max.) again when A/D conversion.

23.2.6 SA-ADC Interrupt Mode Register (SADIMOD)

SADIMOD is a SFR to select the interrupt mode of the SA-ADC.

Address : 0xF808 (SADIMOD)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADIMOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADIMD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	SADIMD0	Select the occurrence timing of SA-ADC interrupt request. 0: Make the interrupt request after the A/D conversion is completed on all channels (Initial value) 1: Make the interrupt request whenever the A/D conversion is completed on each channel

23.2.7 SA-ADC Trigger Register (SADTRG)

SADTRG is a SFR to control the trigger event for the SA-ADC.

Address : 0xF80A (SADTRG)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADTRG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	SASTS2	SASTS1	SASTS0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2 to 0	SASTS2 to SASTS0	Select the source of the A/D conversion start trigger. 000: 16 bit timer 0 interrupt (TM0INT) (initial value) 001: Operational timer 0 trigger (OTM0TRG) 010: Operational timer 1 trigger (OTM1TRG) 011: Operational timer 2 trigger (OTM2TRG) 100: Operational timer 3 trigger (OTM3TRG) 101: Operational timer 4 trigger (OTM4TRG) 110: Operational timer 5 trigger (OTM5TRG) 111: Do not use

23.2.8 SA-ADC Enable Register 0 (SADEN0)

SADEN0 is a SFR to select channels of the A/D converter and enable/disable the conversion.

Address : 0xF80C (SADEN0L/SADEN0), 0xF80D (SADEN0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN0															
Byte	SADEN0H								SADEN0L							
Bit	-	-	-	-	-	-	-	-	-	-	-	SACH 04	SACH 03	SACH 02	SACH 01	SACH 00
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
This bit is used to control enable/disable the conversion on a target channel.
0: Disabled (Initial value)
1: Enabled

When multiple bits of SACHn (n=00 to 18) are set to "1", the A/D conversion starts in the order of smaller channel number.

Bit No.	Bit symbol name	Description (target channel)
15 to 5	-	Reserved bits
4	SACH04	Channel 4
3	SACH03	Channel 3
2	SACH02	Channel 2
1	SACH01	Channel 1
0	SACH00	Channel 0

[Note]
- **Do not start the A/D conversion when the all bits of SACHn (n=00 to 18) are "0". In that case SARUN bit of SADCON register does not get to "1".**

23.2.9 SA-ADC Enable Register 1 (SADEN1)

SADEN1 is a SFR to select channels of the A/D converter and enable/disable the conversion.

Address : 0xF80E (SADEN1L/SADEN1), 0xF80F (SADEN1H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN1															
Byte	SADEN1H								SADEN1L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	SACH 18	SACH 17	SACH 16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description (target channel)
15 to 3	-	Reserved bits
2	SACH18	Control the enable/disable of conversion of Channel 18 (A/D converter test function). 0: Disable the conversion on channel 18 (initial value) 1: Enable the conversion on channel 18
1	SACH17	Control the enable/disable of conversion of Channel 17 (PGA input). 0: Disable the conversion on channel 17 (initial value) 1: Enable the conversion on channel 17
0	SACH16	Control the enable/disable of conversion of Channel 16(temperature sensors). 0: Disable the conversion on channel 16 (initial value) 1: Enable the conversion on channel 16

[Note]

- **Do not start the A/D conversion when the all bits of SACHn (n=00 to 18) are "0". In that case SARUN bit of SADCON register does not get to "1".**
- **When using channel 16 (SACH16), permit the temperature sensor operation by VREFCON register.**
- **When using channel 18 (SACH18), enable the A/D converter test function by SADTMOD register (set to something other than "00").**

23.2.10 SA-ADC Result Register (SADR)

SADR is a read-only SFR to store A/D conversion results of channel 0 to 4 and channel 16 (temperature sensor), channel 17 (PGA), and channel 18 (A/D converter test function).

Address : 0xF83E (SADRL/SADR), 0xF83F (SADRH)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADR															
Byte	SADRH								SADRL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The A/D conversion results of all channels are stored to this register. The result of each channel is overwritten.
The A/D conversion test result on channel 17 is stored to this register only.

Symbol name	Channel
SADR	Latest conversion result of channels 0 to 4 ,16 to 18

[Note]

- **When different clocks for the source clocks of the SAD_CLK and the system clock are selected, wrong value might be readout during the update. Make sure that SARUN bit of SADCON register is "0" before reading. Also, during continuous A/D conversion (SALP=1), read the SADR register twice and make sure that the first and second values match.**

23.2.11 SA-ADC Result Register n (SADRn : n=0 to 4, 16,17)

SADRn is a read-only SFR to store A/D conversion results of channel 0 to 4 and channel 16 (temperature sensor) and channel 17 (PGA).

Address : 0xF840 (SADR0L/SADR0), 0xF841 (SADR0H),
0xF842 (SADR1L/SADR1), 0xF843 (SADR1H),
0xF844 (SADR2L/SADR2), 0xF845 (SADR2H),
0xF846 (SADR3L/SADR3), 0xF847 (SADR3H),
0xF848 (SADR4L/SADR4), 0xF849 (SADR4H),
0xF860 (SADR16L/SADR16), 0xF861 (SADR16H),
0xF862 (SADR17L/SADR17), 0xF863 (SADR17H)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADRn															
Byte	SADRnH								SADRnL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The A/D conversion result of each channel can be read from SADRn.

Symbol name	Channel
SADR0	Channel 0 (AIN0)
SADR1	Channel 1 (AIN1)
SADR2	Channel 2 (AIN2)
SADR3	Channel 3 (AIN3)
SADR4	Channel 4 (AIN4)
SADR16	Channel 16 (Temperature sensor)
SADR17	Channel 17(PGA)

[Note]

- **If the source clock (HSCLK/LSCLK0) setting of the SAD_CLK differs from the source clock (HSCLK/LSCLK0) setting of the system clock, uncertain data during the A/D conversion result update may be read when SADRn register is read. When the source clock settings of the SAD_CLK and the system clock are different, make sure that SARUN bit of SADCON register is "0" before reading. During continuous A/D conversion (SADP bit of SADMOD register is "1"), read SADRn register twice and make sure that the values are the same.**

23.2.12 SA-ADC Test Mode Register (SADTMOD)

SADTMOD is a SFR to control the SA-ADC test function.

Address : 0xF830 (SADTMOD)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADTM1	SADTM0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This function enables to check if the successive approximation type A/D converter and the analog switch work correctly, by performing the A/D conversion for the full scale, zero scale and the internal reference voltage (2.5 V).
The A/D conversion result is stored in the SA-ADC result register (SADR).

Also, the AIN0-4 input level is measured by using a measurement value of internal reference voltage.
For example:
1: Convert at $V_{ADCREF} = V_{DD}$, SACH18=1, SADTM1,0=3, where the result is “a”
2: Convert at $V_{ADCREF} = V_{DD}$, SACHn=1, where the result is =“b”
"a" is the result of the conversion of the ADC internal reference voltage (2.5 V), and AIN selected in SACHn is b/a [V] .

The seriality measurement can be by setting the SACHn and SACH18 at once.
Read the results from SADRn and SADR.

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1 to 0	SADTM1, SADTM0	Set the successive approximation type A/D converter test function. 00: Do not use the A/D converter test function (Initial value) 01: Full scale A/D conversion 10: Zero scale A/D conversion 11: Internal reference voltage (2.5V) A/D conversion

23.3 Description of Operation

23.3.1 Operation of Successive Approximation Type A/D Converter

Figure 23-2 shows a setting example when one-time A/D conversion is performed using channel 1 and 0.

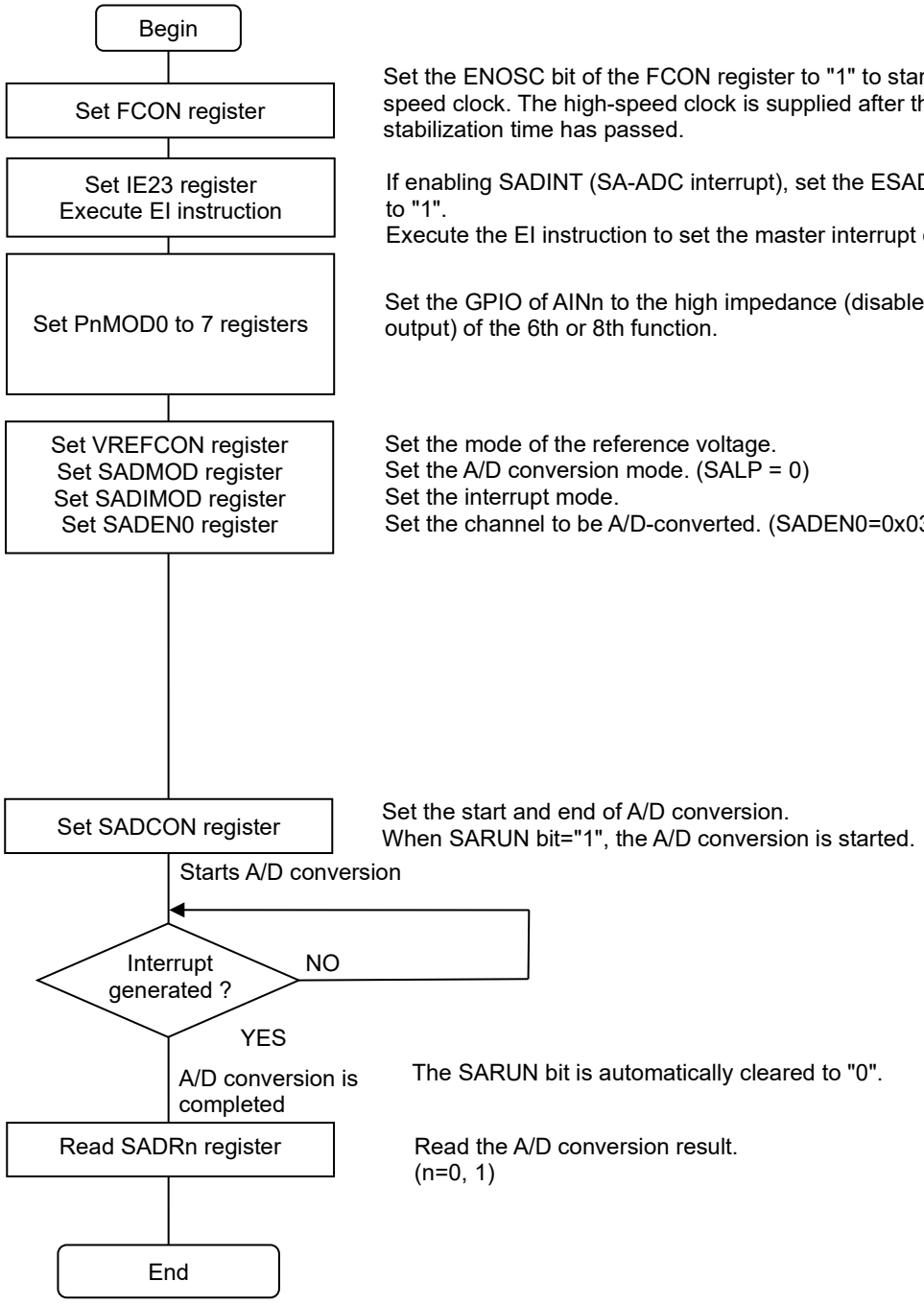


Figure 23-2 Example of A/D Conversion Setting

Figure 23-3 shows a setting example when one-time A/D conversion is performed in HALT mode using channel 1 and 0.

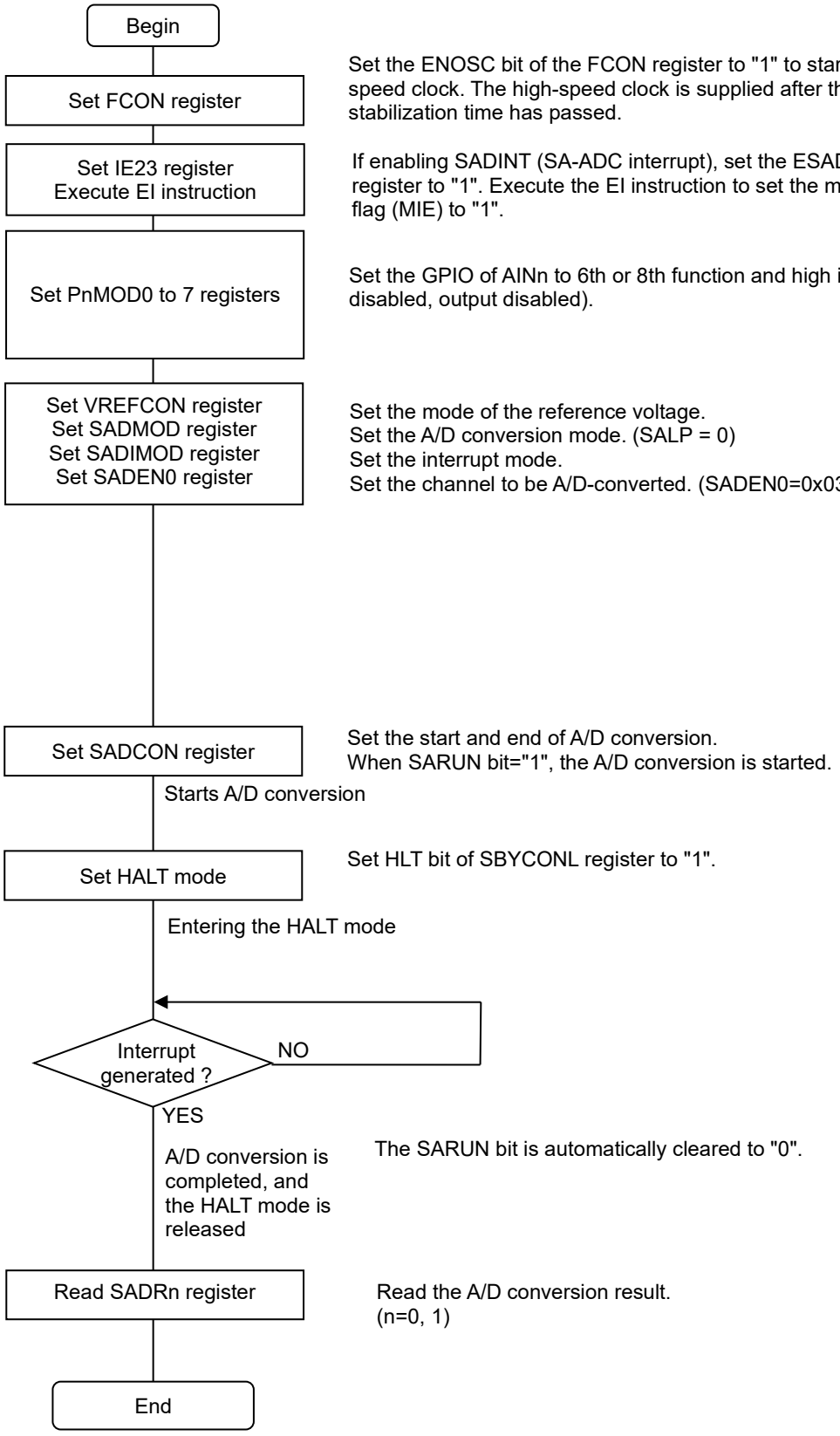


Figure 23-3 Example of A/D Conversion Setting (Converting in HALT mode)

Figure 23-4 shows a setting example when one-time A/D conversion is performed using channel 1 and 0 starting by a trigger event.

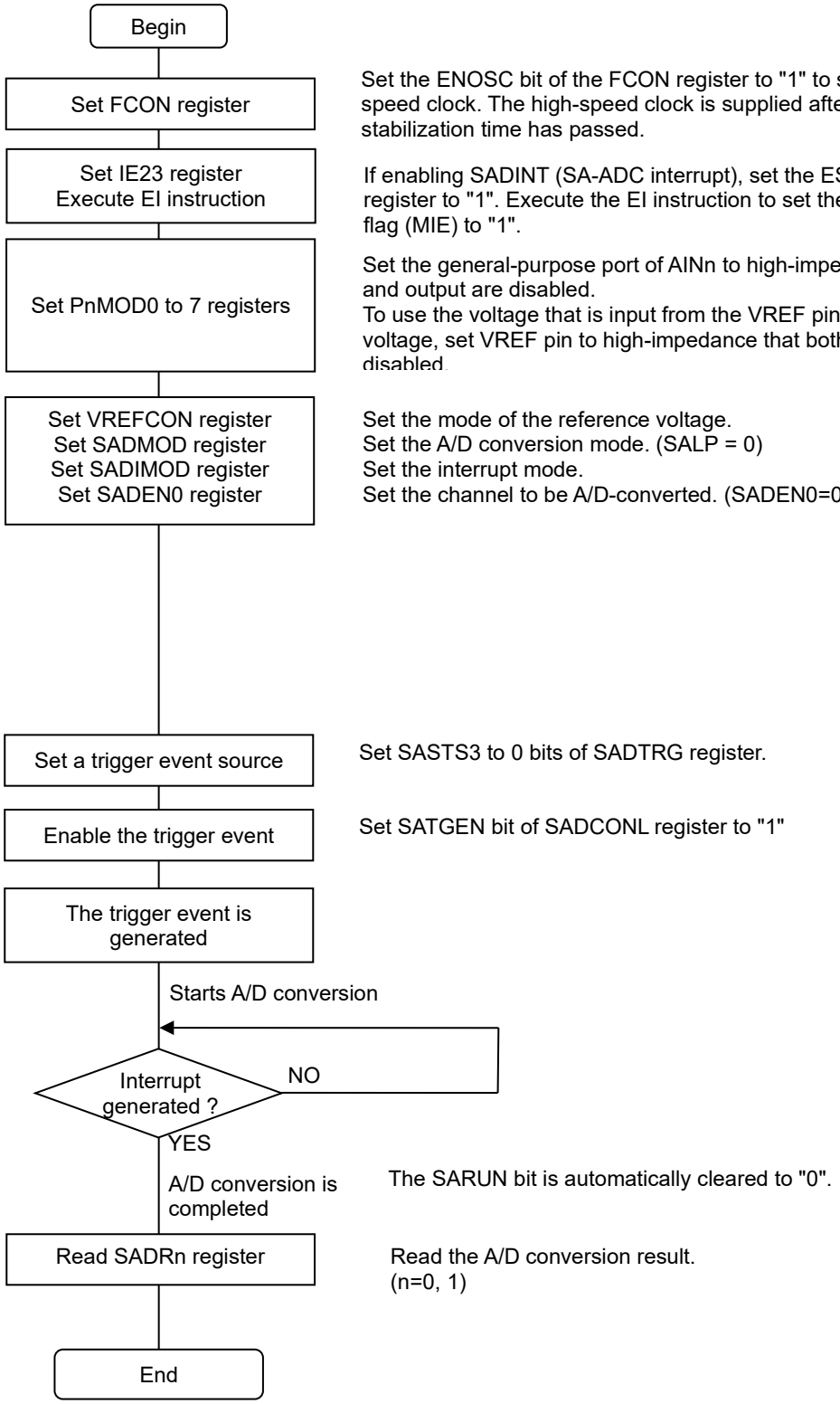
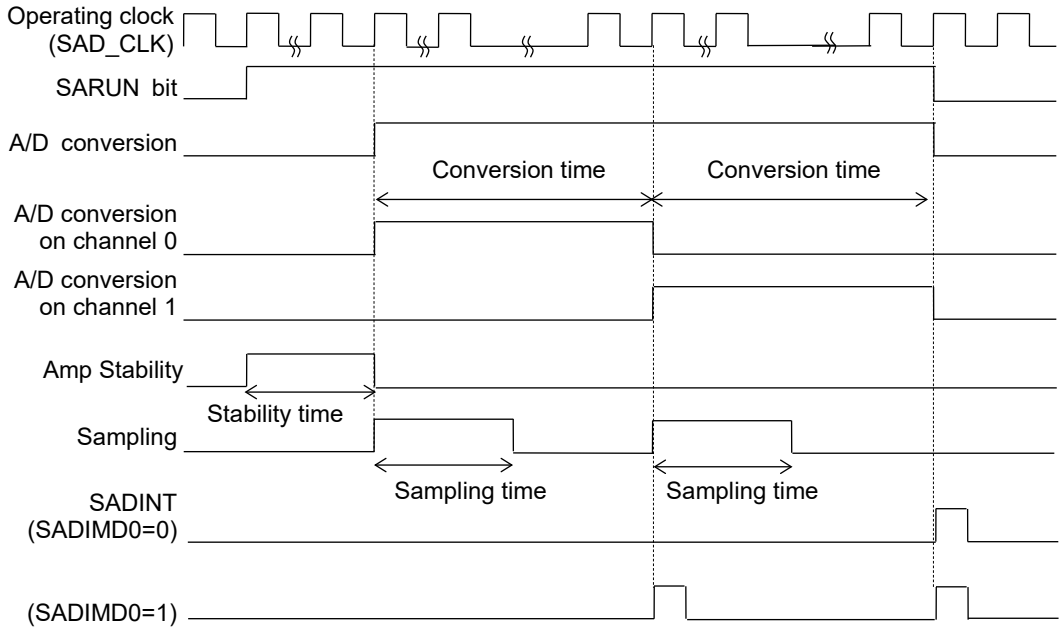


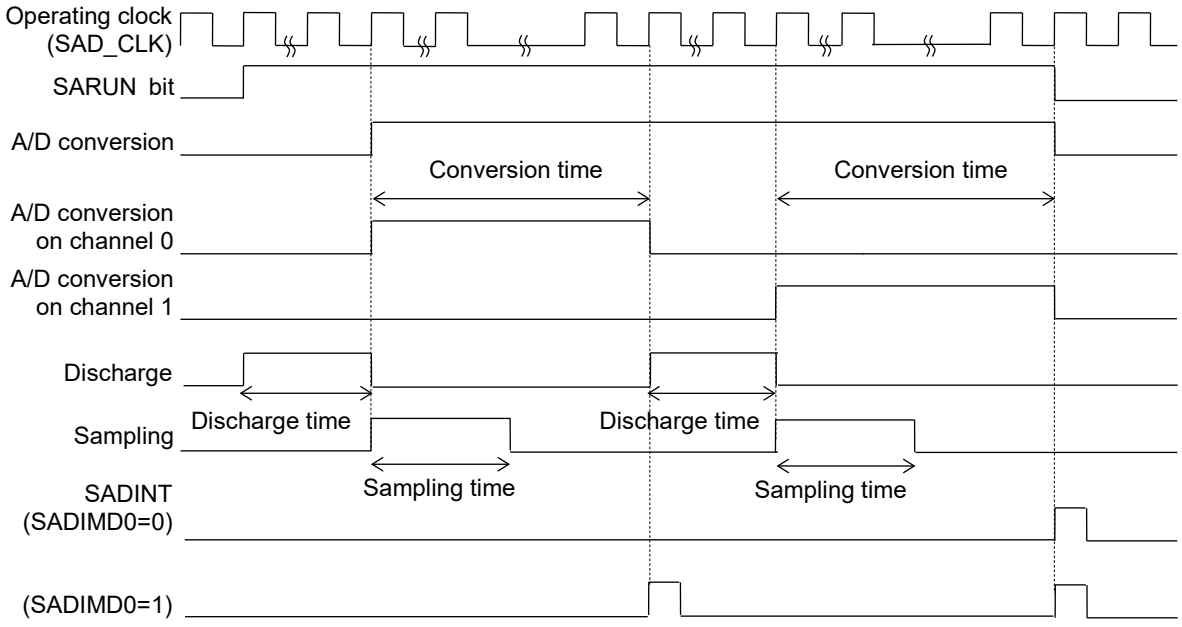
Figure 23-4 Example of A/D Conversion Setting (Start converting by a trigger event)

Figure 23-5 and 23-6 show operation waveforms when one-time A/D conversion is performed using channel 1 and 0.



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Amp stability time is configured by SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT3-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

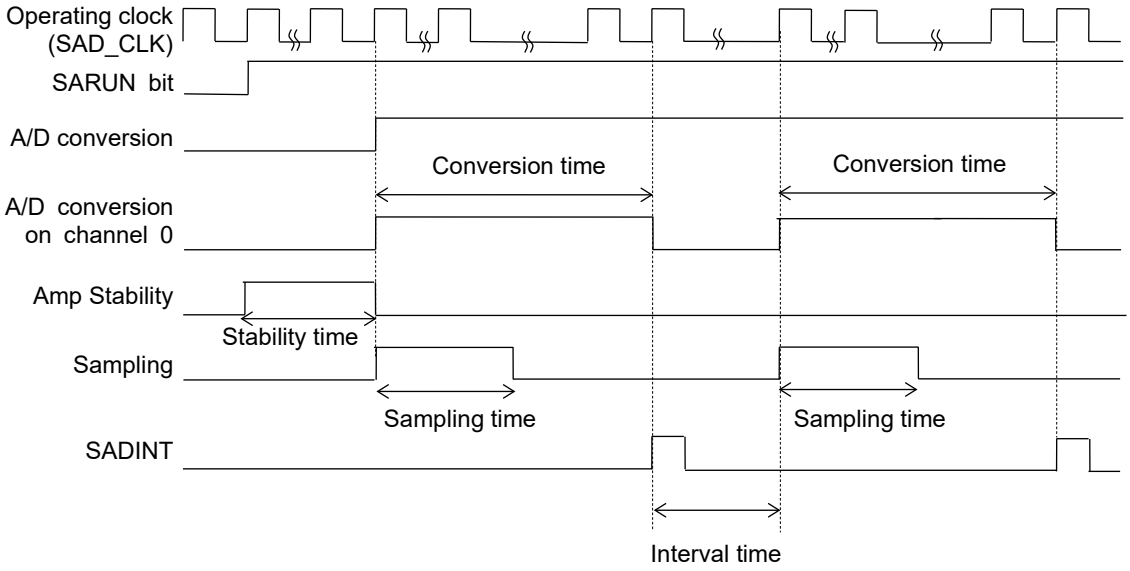
Figure 23-5 Operation Waveforms of A/D Conversion (One-time Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Discharge enabling and time are configured by SAINIT bit and SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT3-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

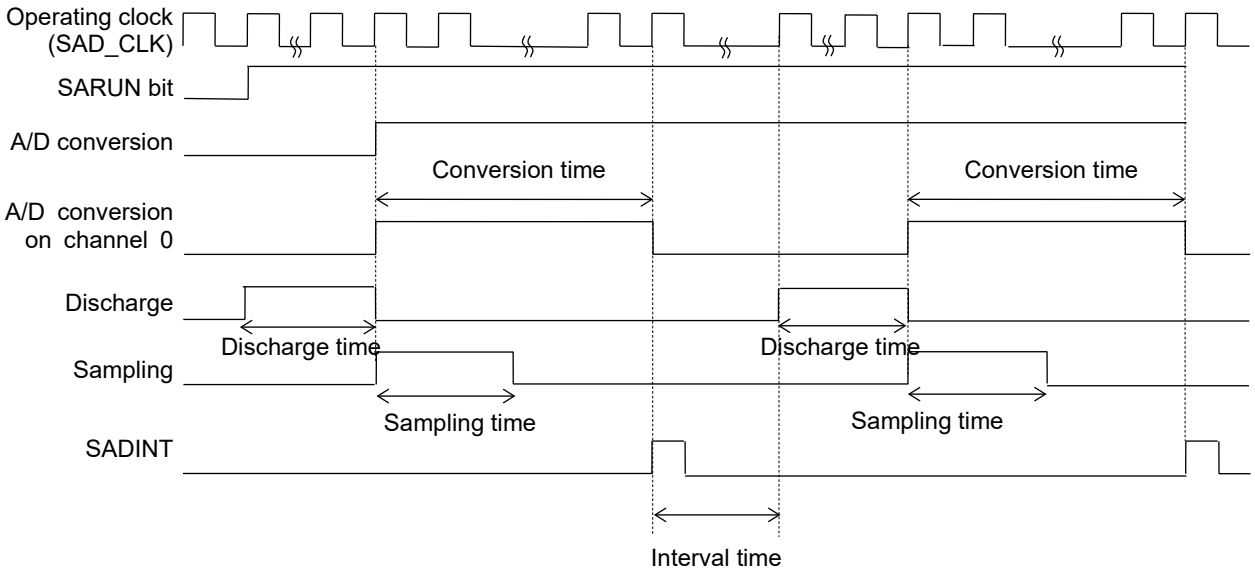
Figure 23-6 Operation Waveforms of A/D Conversion (One-time Conversion, With Discharge)

Figure 23-7 and 23-8 show the operation waveforms when the continuous A/D conversion is performed using channel 0.



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Amp stability time is configured by SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT3-0 bits of SADMOD register.
A/D conversion interval is configured by SADSTM register.

Figure 23-7 Operation Waveforms of A/D Conversion (Continuous Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Discharge enabling and time are configured by SAINIT bit and SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT3-0 bits of SADMOD register.
A/D conversion interval is configured by SADSTM register.

Figure 23-8 Operation Waveforms of A/D Conversion (Continuous Conversion, With Discharge)

23.3.2 How to test the Successive Approximation Type A/D Converter

Full-scale, zero-scale, and ADC internal reference voltage (2.5 V) can be A/D converted for self-testing.
Follow this procedure to check if the successive approximation type A/D converter works correctly. (n=0 to 4)

(1) A/D convert AINn pin. (conversion result 1)
(2) A/D convert AIN=full scale by setting the SADTMOD register (SADTMOD=0x01).
(3) A/D convert the AINn pin. (conversion result 2)
(4) A/D convert AIN=zero scale by setting the SADTMOD register (SADTMOD=0x02).
(5) A/D convert the AINn pin. (conversion result 3)
(6) A/D convert AIN=internal reference voltage(2.5V) by setting the SADTMOD register (SADTMOD=0x03).
(7) A/D convert the AINn pin. (conversion result 4)
(8) Confirm conversion result 1 = conversion result 2 = conversion result 3 = conversion result 4.
Use the same AINn pin for the A/D conversion in (1), (3), (4) and (7).
(9) Confirm the conversion result in (2), (4) and (6) is different each other and also different from the result in (1), (3), (5) and (7).

23.3.3 PGA in Successive Approximation A/D Converter/Temperature Conversion Function

Figure 23-9 shows a setting example when one-time A/D conversion is performed using channel 16(temperature sensor) and channel 17(PGA).

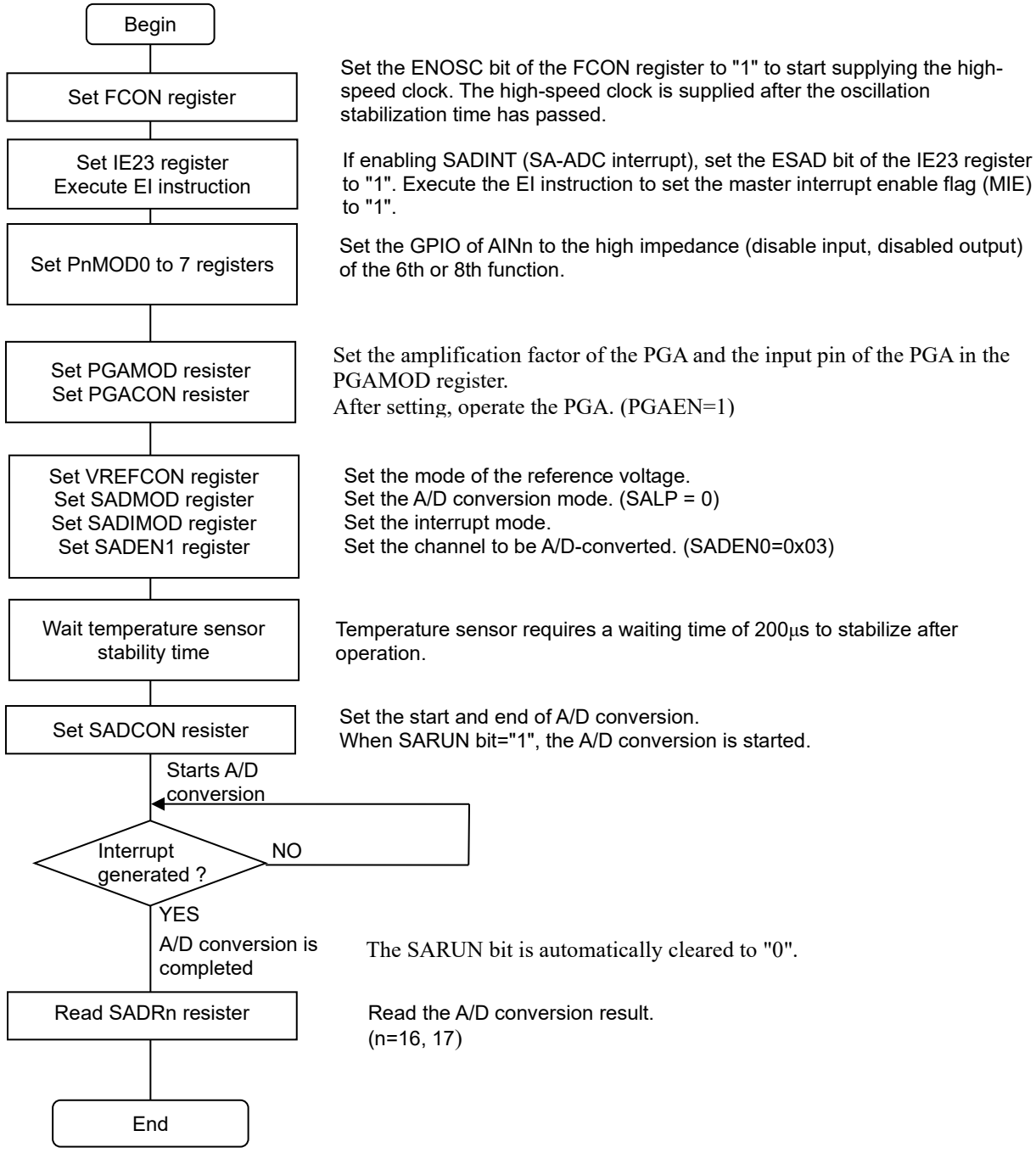


Figure 23-9 Example of A/D Conversion Setting

23.4 Notes on SA-ADC

23.4.1 Sampling Time Setting

Sampling time of the SA-ADC should satisfy the following formula:

$$\text{Sampling time} > 9(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

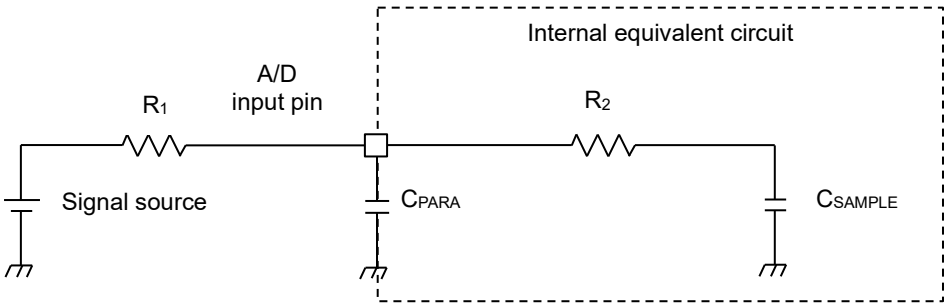
To calculate sampling time more precisely, use the following formula:

$$\text{Sampling time} = \left\{log_e(2^n) + log_e\left(\frac{C_{SAMPLE}}{C_{SAMPLE} + C_{PARA}}\right)\right\}(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

C_{PARA} varies depending on board-layout and connected parts. Please check the accuracy of SA-ADC with the actual board.

R_1 : Input impedance of external resistor
R_2 : Internal resistor value which is the sum of the internal resistor and the ON register of the switch
C_{SAMPLE} : Sample hold capacitor
C_{PARA} : Parasitic capacitance of the A/D input line. (Measure the capacitance between the A/D input line and V_{SS}.)
n : Resolution of SA-ADC

The following diagram shows the equivalent circuit in this case:



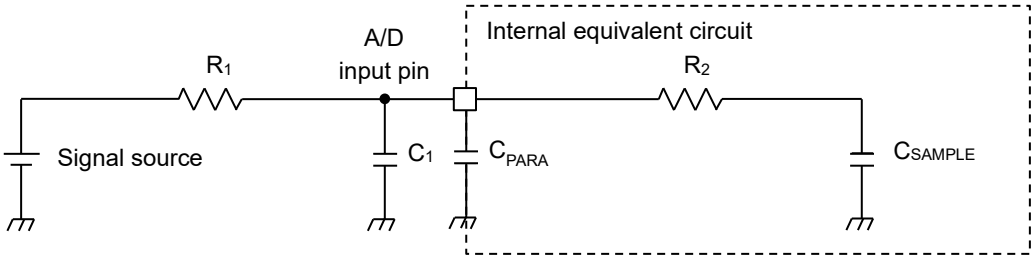
V_{ADCREF}	R_2[kΩ]	C_{SAMPLE}[pF]
$V_{ADCREF}=V_{REFI}$	20	5
$4.5V \leqq V_{ADCREF} \leqq 5.5V$ *VDD reference	10	5

The values above are reference values.
Set the sampling time for V_{REF} condition that includes the lowest voltage of the usage range of V_{REF}.
If the sampling time above is unsatisfied, connect the external capacitor near by A/D input pin to satisfy the following formula.

$$(C_1 + C_{PARA}) > 2^n C_{SAMPLE}$$
$$\text{Sampling time} > 9C_{SAMPLE}R_2$$

C_1 : External capacitor

The equivalent circuit when the external capacitor C_1 is connected is as follows:



Note that the voltage at the A/D input pin transitionally changes due to the external capacitor C_1 and the external resistor R_1. Therefore, when sampling data, wait until the voltage is stabilized. If the stabilization timing is unknown, perform A/D conversion once, then wait for time constant τ ($=R_1C_1$) to 4τ or so and perform A/D conversion again. Confirm that the difference between values is small, and then sample data.

23.4.2 Noise Suppression

In order to prevent deterioration in accuracy of A/D conversion, operate the A/D converter in the environment with little noise.
The following processes are recommended for noise reduction:

- Perform A/D conversion in the HALT mode.
- Do not have clock input/output to and from a pin located in the vicinity of the pin in which A/D conversion is in progress.
- Do not have clock input/output to and from the pin in which A/D conversion is in progress and other A/D conversion pins.

In addition, the capacitor for noise suppression should be connected between VREF and VSS, as well as between VDD and VSS. When connecting, place the capacitor in the immediate vicinity of LSI using short wiring.

Chapter 24 PGA

24. PGA

24.1 General description

This LSI has the PGA (Programable Gain Amplifer) which amplify the input of successive approximation A/D converters.
Table 24-1 shows PGA channel of each package products.

Table 24-1 PGA channels

Channel No.(n)	Availability	
	20 pin Product	24pin Product
0	●	●

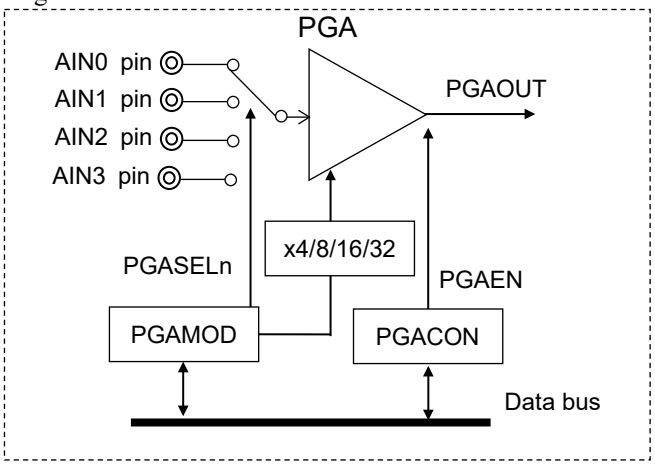
●: Available −: No avaailable

24.1.1 Features

- Amplification factor is selectable from x4, x8, x16, and x32.
- PGA input pin is selectable from AIN0 to AIN3 pin.

24.1.2 Configuration

Fig. 24-1 shows PGA configuration



PGACON : PGA control register
PGAMOD : PGA mode register

Fig. 24-1 PGA circuit configuration

24.1.3 Pin list

PGA input pins are shared with AINn pins.

Pin name	I/O	Function
AINn	I	PGA input (AIN input)

Table 24-2 shows General ports and register setteing used for PGA function. PGA

Table 24-2 General ports and register setteing used for PGA function

Pin name	Shared port		Setting Register	Setting Value	20pin Product	24pin Product
AIN0	P14	6th, 8th functions	P1MOD4	0x50 / 0x70	●	●
AIN1	P15		P1MOD5		●	●
AIN2	P16		P1MOD6		●	●
AIN3	P17		P1MOD7		●	●

●: Available -: Not available

[Note]

- **When PGA is used, following the setting below.**
  [1] Set PnmMD2 to 0bit of the corresponding port n-mode register m (n: 0 to 2, m: 0 to 7) to the 6th or 8th function,
  [2] Write "0" to the corresponding bits of input permission (PnmIE) and output permission (PnmOE).
  [3] set the general-purpose port to high impedance.
  In other settings, through current may occurs.
- **During PGA is in operation, the influence of the PGA output voltage fluctuates due to the noise can be reduced by not switching adjacent pins on AINn.**

24.2 Register description

24.2.1 List of register

Address	Name	Symbol		R/W	Sice	Initial value
		Byte	Word			
0xF8B0	PGA control register	PGACON	-	R/W	8	0x00
0xF8B1	Reserved register	-	-	-	-	-
0xF8B2	PGA mode register	PGAMOD	-	R/W	8	0x00
0xF8B3	Reserved register	-	-	-	-	-

24.2.2 PGA Control Register (PGACON)

PGACON is a SFR to control PGA.

Address: 0xF8B0（PGACON）
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								PGACON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PGAEN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Desription
7 to 1	-	Reserved bit
0	PGAEN	Control PGA operation permition/stop 0: Stop PGA operation (Initial value) 1: Permit PGA operation

[Note]

- **When PGA is used, following the setting below.**
  [1] Set PnmMD2 to 0bit of the corresponding port n-mode register m (n: 0 to 2, m: 0 to 7) to the 6th or 8th function,
  [2] Write "0" to the corresponding bits of input permission (PnmIE) and output permission (PnmOE).
  [3] set the general-purpose port to high impedance.
  In other settings, through current may occurs.
- **After setting the PGAEN bit to "1", it takes 200µs for PGA to stabilize. A/D conversion should be started after this stabilization time has elapsed.**
  After the stabilization time (200µs) has elapsed, PGA remains stable until the operation stops, so there is no need to wait for 200µs again for each A/D conversions.

24.2.3 PGA Mode register (PGAMOD)

PGAMOD is a SFR to set PGA amplification factor and input pin.
Write to PGAMOD register while PGA operation is stopped (PGAEN bit = "0" in PGACON register).

Address: 0xF8B2(PGAMOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	—															
Byte	—								PGAMOD							
Bit	-	-	-	-	-	-	-	-	-	-	PGASEL1	PGASEL0	-	-	PGAGAIN1	PGAGAIN0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Desription
7 to 6	-	Reserved bit
5 to 4	PGASEL1 to PGASEL0	Select PGA input pin 00: AIN0 (Initial vaule) 01: AIN1 10: AIN2 11: AIN3
3 to 2	-	Reserved bit
1 to 0	PGAGAIN1 to PGAGAIN0	Set PGA Amplification factor 00: x4 (Initial value) 01: x8 10: x16 11: x32

[Note]

- **Writing to the PGAMOD register while PGA operation is permited, the operation is not guaranteed.**

24.3 Features

24.3.1 PGA Operation

Fig. 24-2 shows the flowchart of A/D convert by using PGA operation.
This example is shown when A/D conversion is performed only once using channel 17 (PGA output) and the amplification factor is set to x16.

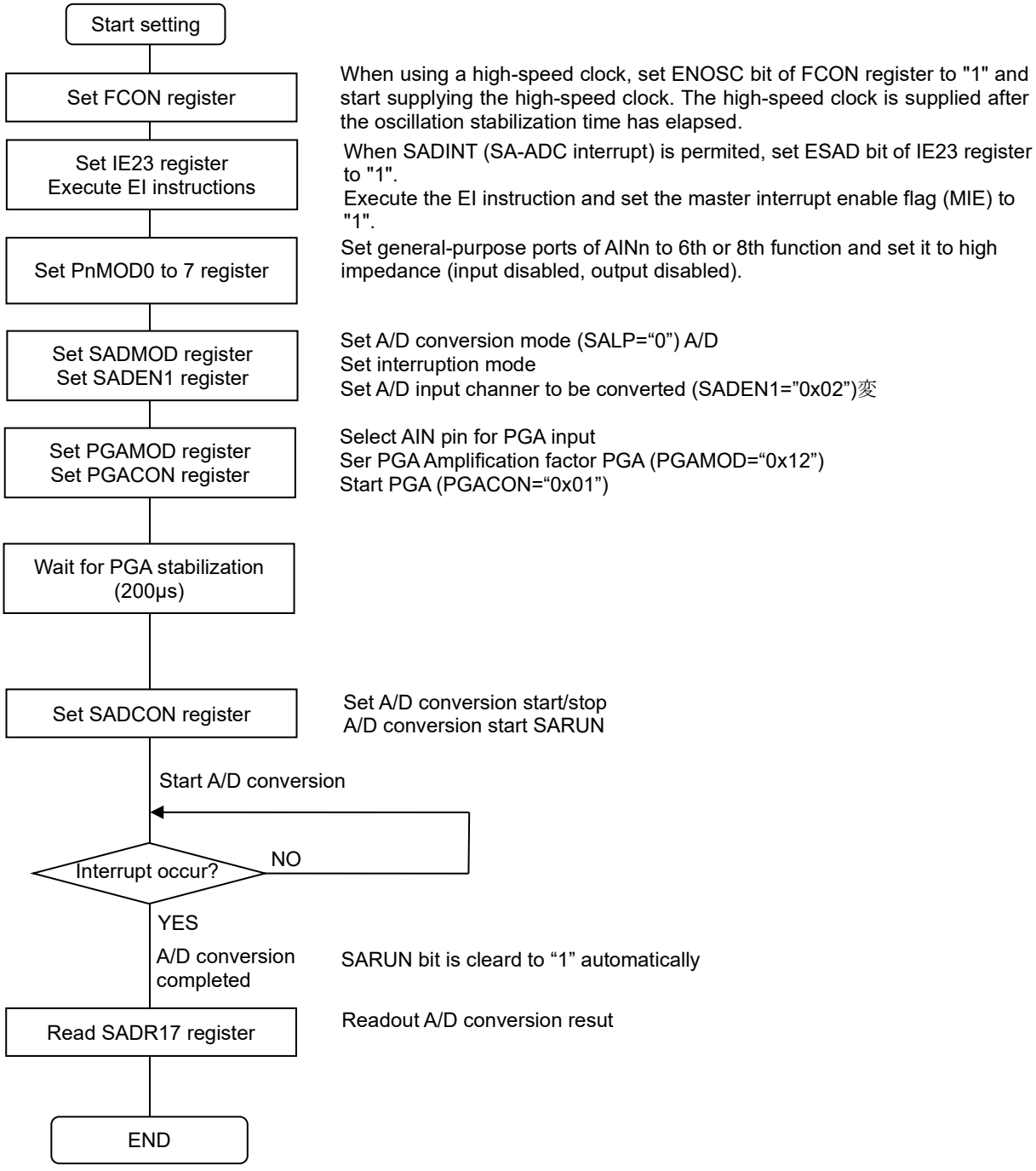


Fig. 24-2 Flowchart of A/D Conversion using PGA

Chapter 26 Flash Memory

26. Flash Memory

26.1 General Description

This LSI has the flash memory in the program memory space and data flash area. For details of the program memory space and data flash area, see Chapter 2 "CPU and Memory Space".
The flash memory is programmable by following three ways.

- The ways of programming the flash memory

Programming method	Tool/Register/Communication	Reference Chapter
Programming by the on-chip debug function	On-chip debug emulator or other flash programmers	Chapter 28 "On chip Debug function"
Self-Programming by using the special function register (SFR)	Special Function Registers (SFRs) for programming the flash memory	Section 26.3 "Self-programming"
Programming by the ISP (In-System Programming) function	UART communication with an external device 3rd Party Flash programmers (*1)	Section 26.4 "In-System Programming function"

*1: Contact the 3rd party makers for details about the Flash programmer.

The specification of the program memory space and data flash are is dependent of the product.

- Program memory space and Data flash area Overview (Size and Address)

Part name	Program memory space		Data flash area	
	Size	Address	Size	Address
ML62Q2033/2043	16KByte	0x0:0000 to 0x0:3FFF	4KByte (128Byte x 32sector)	0x1F:0000 to 0x1F:0FFF
ML62Q2035/2045	32KByte	0x0:0000 to 0x0:7FFF		

- Program memory space and Data flash area Overview (Functions and Characteristics)

Item		Program memory space	Data flash area
Erasing and programming unit	Chip erasing (ISP only)	All area	All area
	Block erasing	16K byte	all area
	Sector erasing	1K byte	128 byte
	Programming	SFR:2 byte (16bit) ISP:4 byte (32bit)	1 byte (8bit)
Erasing and programming time	Chip erasing (ISP only)	Max. 50ms	Max. 50ms
	Block erasing		
	Sector erasing		
	Programming	SFR:max 60µs ISP:max 160µs	Max. 40µs
Programming cycle		100 times	10,000 times
Erasing and programming temperature		0°C to +40°C	-40°C to +85°C
Background operation(BGO) function		-	Yes
Erasing and programming completion Interrupt		No	Yes

26.1.1 List of Pins

Programming by the ISP function uses the following pins.

Pin name	I/O	Function
RESET_N	I	Signal input for ISP
TEST0	I/O	Signal input for ISP and data input/output for UART

26.2 Register Description

26.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF090	Flash address register	FLASHAL	FLASHA	R/W	8/16	0xFF
0xF091		FLASHAH		R/W	8	0xFF
0xF092	Flash data register 0	FLASHD0L	FLASHD0	R/W	8/16	0xFF
0xF093		FLASHD0H		R/W	8	0xFF
0xF094	Reserved register	-	-	-	-	-
0xF095	Reserved register	-	-	-	-	-
0xF096	Flash control register	FLASHCON	-	W	8	0x00
0xF097	Reserved register	-	-	-	-	-
0xF098	Flash acceptor	FLASHACP	-	W	8	0x00
0xF099	Reserved register	-	-	-	-	-
0xF09A	Flash segment register	FLASHSEG	-	R/W	8	0x10
0xF09B	Reserved register	-	-	-	-	-
0xF09C	Flash self register	FLASHSLF	-	R/W	8	0x00
0xF09D	Reserved register	-	-	-	-	-
0xF09E	Flash status register	FLASHSTA	-	R	8	0x00
0xF09F	Reserved register	-	-	-	-	-

26.2.2 Flash Address Register (FLASHA)

FLASHA is a SFR to set the erasing and programming address.

Address : 0xF090 (FLASHAL/FLASHA), 0xF091 (FLASHAH)
Access : R/W
Access size : 8/16 bit
Initial value : 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHA															
Byte	FLASHAH								FLASHAL							
Bit	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 0	FA15 to FA0	Set the erasing or programming address.

[Note]

- **Note that programming for the program memory space is performed by the unit of 2 bytes. Because of this, the setting values in the FA0 bit are ignored.**

26.2.3 Flash Segment Register (FLASHSEG)

FLASHSEG is a SFR used to set the segment for erasing and programming the flash memory.

Address : 0xF09A (FLASHSEG)
Access : R/W
Access size : 8 bit
Initial value : 0x10

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHSEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	FSEG4	FSEG3	FSEG2	FSEG1	FSEG0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 5	-	Reserved bits
4 to 0	FSEG4 to FSEG0	Set the flash memory segment address.

Table 26-1 shows the address setting value for block erasing and Table 26-2 shows the address setting value for sector erasing.

Table 26-1 Address Setting Values for Block Erasing

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Block 0	0x0000 to 0x3FFF	16KByte	0x00	0x0000
	Block 1	0x4000 to 0x7FFF	16KByte		0x4000
Segment 31	Block 0	0x0000 to 0x0FFF	4KByte	0x1F	0x0000

Table 26-2 Address Setting Values for Sector Erasing

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Sector0	0x0000 to 0x03FF	1KByte	0x00	0x0000
	Sector1	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector30	0x7800 to 0x7BFF	1KByte		0x7800
	Sector31	0x7C00 to 0x7FFF	1KByte		0x7C00
Segment 31	Sector0	0x0000 to 0x007F	128Byte	0x1F	0x0000
	Sector1	0x0080 to 0x00FF	128Byte		0x0080
	Sector2	0x0100 to 0x017F	128Byte		0x0100
	Sector3	0x0180 to 0x01FF	128Byte		0x0180
	:	:	:		:
	Sector12	0x0600 to 0x067F	128Byte		0x0600
	Sector13	0x0680 to 0x06FF	128Byte		0x0680
	Sector14	0x0700 to 0x077F	128Byte		0x0700
	Sector15	0x0780 to 0x07FF	128Byte		0x0780
	:	:	:		:
	Sector28	0x0E00 to 0x0E7F	128Byte		0x0E00
	Sector29	0x0E80 to 0x0EFF	128Byte		0x0E80
	Sector30	0x0F00 to 0x0F7F	128Byte		0x0F00
	Sector31	0x0F80 to 0x0FFF	128Byte		0x0F80

26.2.4 Flash Data Register 0 (FLASHD0)

FLASHD0 is a SFR used to set programming data.

Address : 0xF092 (FLASHD0L/FLASHD0), 0xF093 (FLASHD0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHD0															
Byte	FLASHD0H								FLASHD0L							
Bit	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	FD15 to FD8	Set the 2nd byte data.
7 to 0	FD7 to FD0	Set the 1st byte data.

There are some differences for programming the program memory space and the data flash area.

Programming target	Register	How to start programming to flash	Description
Program memory space	2 bytes specified in FLASHD0 register	Writing data into FLASHD0/FLASHD0H	
Data flash area	FLASHD0L register only (1 byte) in FLASHD0 register	Writing data into FLASHD0/FLASHD0L	Data written into FLASHD0H of FLASHD0 resister are invalid.

[Note]

- **Specify a segment address to FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.**
- **During programming data-flash, a CPU can execute instruction by the background operation function; BGO. Confirm FDPRSTA bit of FLASHSTA register for complition of programming.**
- **Erase data in the addresses in advance. Programmed data without erase is unguaranteed.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

26.2.5 Flash Control Register (FLASHCON)

FLASHCON is a write-only SFR used to control the block erasing and sector erasing for the flash memory. This register always returns 0x00 for reading.

Address : 0xF096 (FLASHCON)
Access : W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSERS	FERS
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1, 0	FSERS, FERS	Start the sector erasing or block erasing. Setting the FSERS bit to "1" starts erasing the sector and setting the FERS bit to "1" starts erasing the block specified by the FLASHSEG and FLASHAH register. 00: No function (Initial value) 01: Start block erasing 10: Start sector erasing 11: Do not use (No function)

26.2.6 Flash Acceptor (FLASHACP)

FLASHACP is a write-only SFR used to accept for erasing/programming the flash memory.

Address : 0xF098 (FLASHACP)
Access : W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHACP							
Bit	-	-	-	-	-	-	-	-	fac7	fac6	fac5	fac4	fac3	fac2	fac1	fac0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These bits are used to accept for erasing/programming the flash memory in order to prevent an unintended erasing/programming operation.
When "0xFA" and "0xF5" are written to the FLASHACP in this order, the erasing or programming function is enabled only once. For subsequent erasing or programming, "0xFA" and "0xF5" must be written to FLASHACP each time.

Even if other instructions are executed between the instruction that writes "0xFA" and "0xF5" to the FLASHACP, the erasing or programming function is still valid.
If data other than "0xF5" is written to the FLASHACP after "0xFA" is written, "0xFA" becomes invalid. In this case, it needs to write "0xFA" again.

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

26.2.7 Flash Self Register (FLASHSLF)

FLASHSLF is a SFR used to enable erasing and programming the flash memory. When system clock is the low-speed clock, it is not writable.

Address : 0xF09C (FLASHSLF)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHSLF							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSELF
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	FSELF	Enable erasing and programming the flash memory. This bit is kept after completed erasing/programming. 0: Disabled (Initial value) 1: Enabled

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

26.2.8 Flash Status Register (FLASHSTA)

FLASHSTA is a read-only SFR used to check status of the flash memory.

Address : 0xF09E (FLASHSTA)
Access : R
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHSTA							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FDPRSTA	FDERSTA
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	FDPRSTA	Indicate whether the data flash area is in the state of programming. 0: Not in the state of programming (Initial value) 1: In the state of programming
0	FDERSTA	Indicate whether the data flash area is in the state of erasing. 0: Not in the state of erasing (Initial value) 1: In the state of erasing

This register is used when erasing or programming the data flash memory.

Erasing/Programming target	Availability to read this register while erasing/programming	Description
Program memory space	Not available	Do not use the FLASHSTA register.
Data flash area	Available	Start erasing/programming the flash checking the bit is "0".

The CPU stops running the program codes while erasing or programming the program flash memory, therefore FLASHSTA is not readable in that case.

As the Back Ground Operation (BGO) function allows the CPU continue running the program codes, make a process for the next erasing and programming by checking the FDERSTA bit or FDPRSTA bit to see if the erasing or programming the data flash memory is completed.

[Note]

- **Perform the erasing or programming after checking FDERSTA bit or FDPRSTA bit are "0". Do not perform the erasing or programming when either FDERSTA bit or FDPRSTA bit is "1".**

26.3 Self-programming

The self-programming is the function to program (erase and program) the program memory space and data flash area using SFRs.
Table 26-3 shows the self-programming specifications for each of the program memory space and data flash area.

Table 26-3 Self-programming of Program Memory Space and Data Flash Area

		Program memory space (Segment 0)	Data flash area (segment 31)
Programming unit	Erasing block	16 Kbyte	all area
	Erasing sector	1 Kbyte	128 byte
	Programming	2 byte	1 byte
CPU operation during block/sector erase or program		Stop program processing (after completion of erasing/programming, resume program processing)	Continue program processing through the background operation (BGO) function
Confirmation of end of block/sector erasing or programming		Confirmation not required (as program run is stopped during erasing/programming)	Confirmation can be made through FLASHSTA register
Target area where block/sector erasing has been applied		Every bit becomes "1" (the bit written with "0" by writing becomes "0" from "1")	
Note on data programming		Erase the area to be reprogrammed (data programmed without erasing is unguaranteed)	
Function to prevent unintended erasing/programming		Flash self-register (FLASHSLF) and flash acceptor (FLASHACP) incorporated (*1)	
Flash memory erasing/programming		Supported only when system clock is the high-speed clock (*2)	
Note on user program programming		Before programming the user program, prepare a program for self-programming in the program code area which is not erased/reprogrammed	-
Remapping function		User program update, etc. can be performed by simultaneously using remapping function	-

*1: After the programming is enabled by the FLASHSLF register, if "0xFA" and "0xF5" are written to the flash acceptor (FLASHACP), block/sector erase or reprogram is enabled only once.
*2: See Chapter 6 "Clock Generation Circuit" for enabling oscillation of the high-speed oscillation circuit and switching the system clock.

26.3.1 Notes on Debugging Self-programming Code

When debugging the area within the scope of program for self-programming (from setting the flash acceptor to writing the flash data register 0) using U16 development environment (debugger), use the debugger according to the precautions described in Table 26-4.

Table 26-4 Notes on Debugging Self-programming

Limited function	Notes
Breakpoint setting	Do not perform the real time execution with break points set in the scope of program for self-programming (from setting the flash acceptor to setting the flash data register0). Otherwise, the flash memory may not be reprogrammed if break points occur within the scope of program for self-programming.
Step execution	Do not perform the step execution within the scope of program for self-programming. Otherwise, the flash memory may not be reprogrammed if the step execution is performed within the scope of program for self-programming.

26.3.2 Programming Program Memory Space

In the program memory space (flash memory), block erase in units of 16 Kbytes, sector erase in units of 1 Kbyte, and reprogram in units of 2 bytes can be executed.
Figure 26-1 shows the flow diagram for erasing the program memory space.

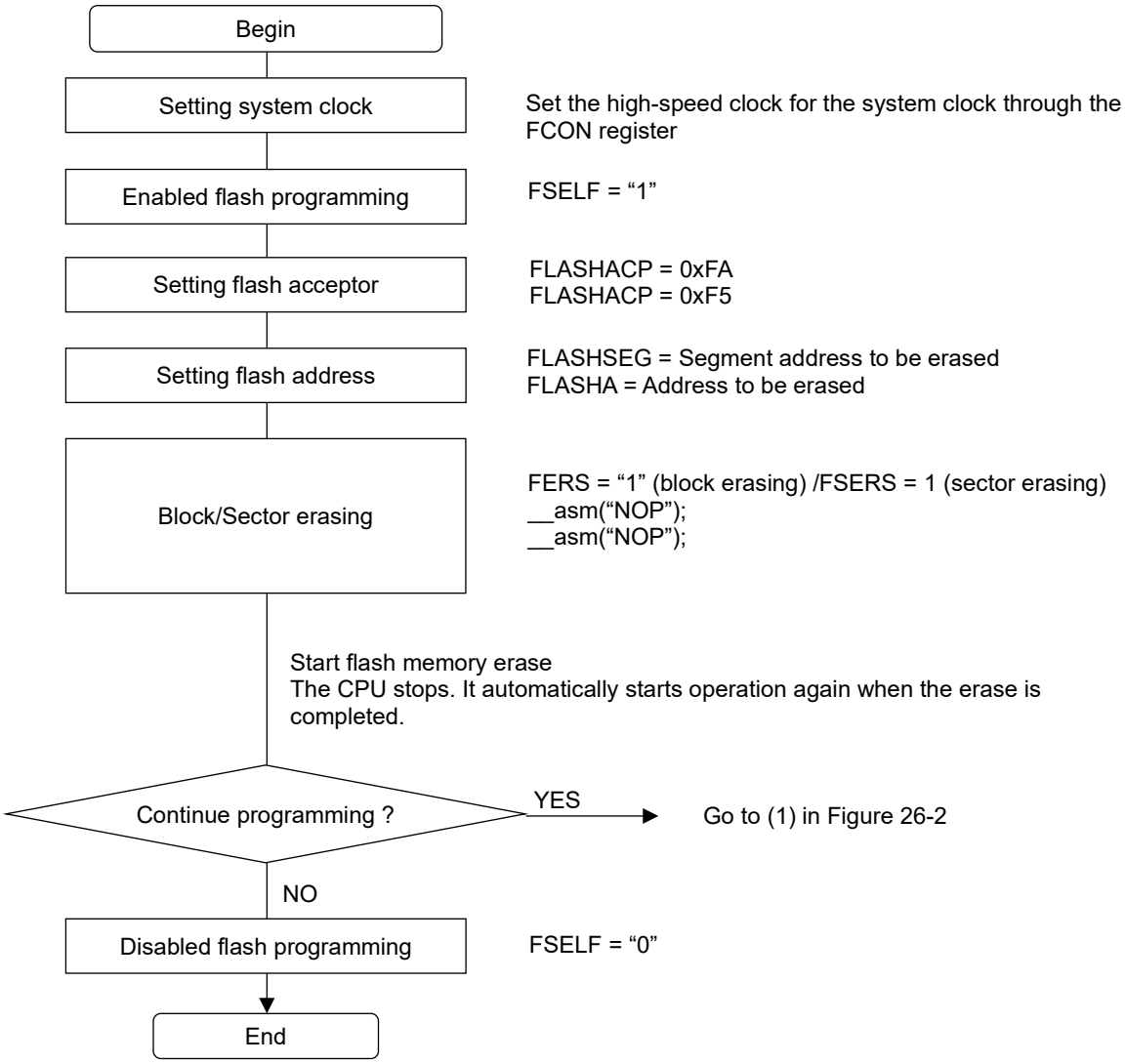


Figure 26-1 Flow Diagram for Erasing Program Memory Space

[Note]

- **Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.**
- **During block/sector erasing, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of FLASHCON register to "1".**

Figure 26-2 shows the flow diagram for programming the program memory space.

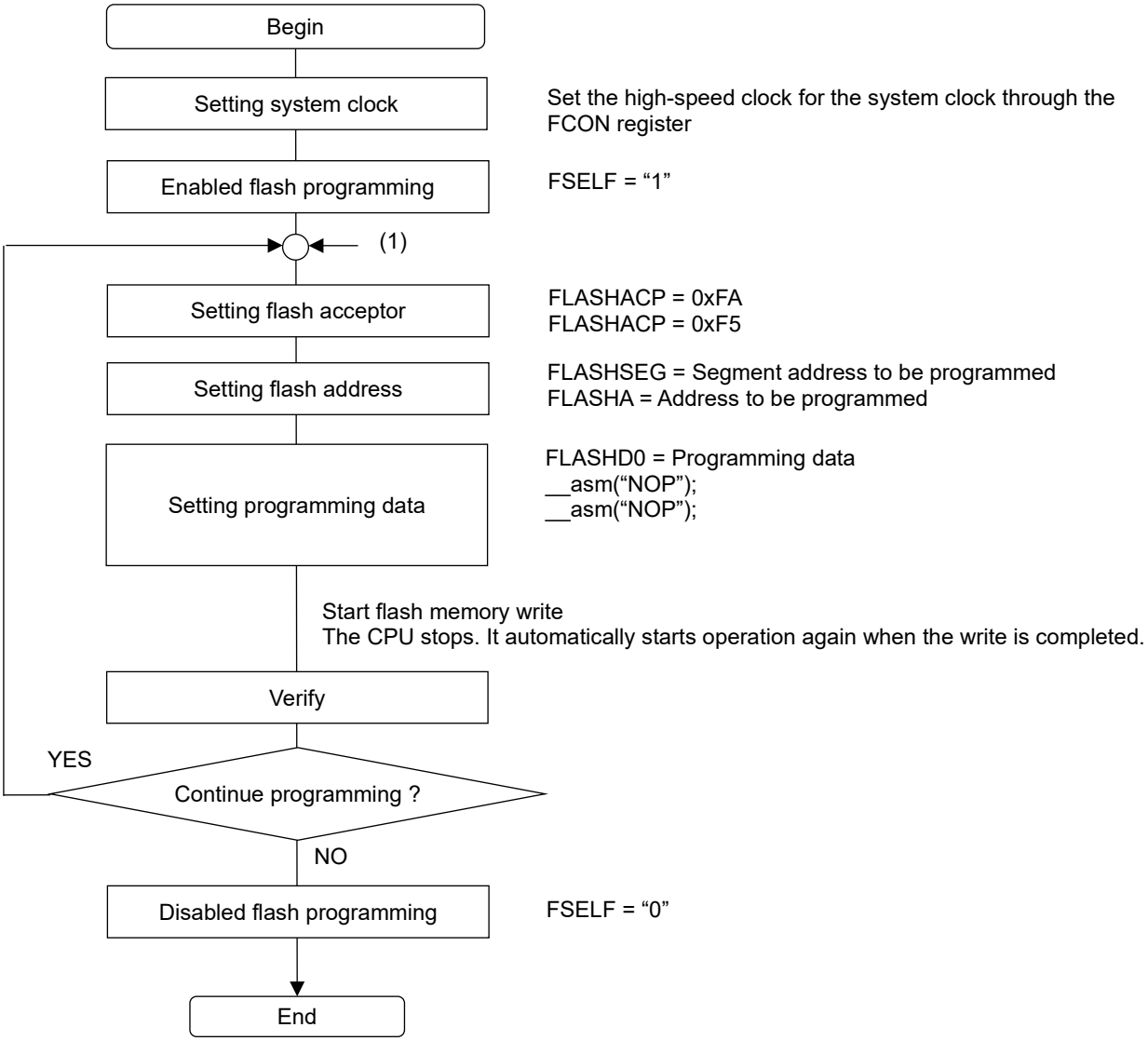


Figure 26-2 Flow Diagram for Programming Program Memory Space

[Note]

- **Only write areas irrelevant to program processing. If programming the area where program processing is in progress, the LSI works incorrectly.**
- **During the programming, the CPU stops the operation for maximum 40 µs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For data programming setting, place two NOP instructions following the instruction used to set the programming data in FLASHD0 register.**

26.3.3 Programming Data Flash Area

In the data flash area (flash memory), block erase in all area, sector erase in units of 128 bytes, and programming in units of 1 byte can be executed. During block/sector erase or program in the data flash area, the CPU continues program processing using the background operation (BGO) function.
Figure 26-3 shows the flow diagram for erasing the data flash area.

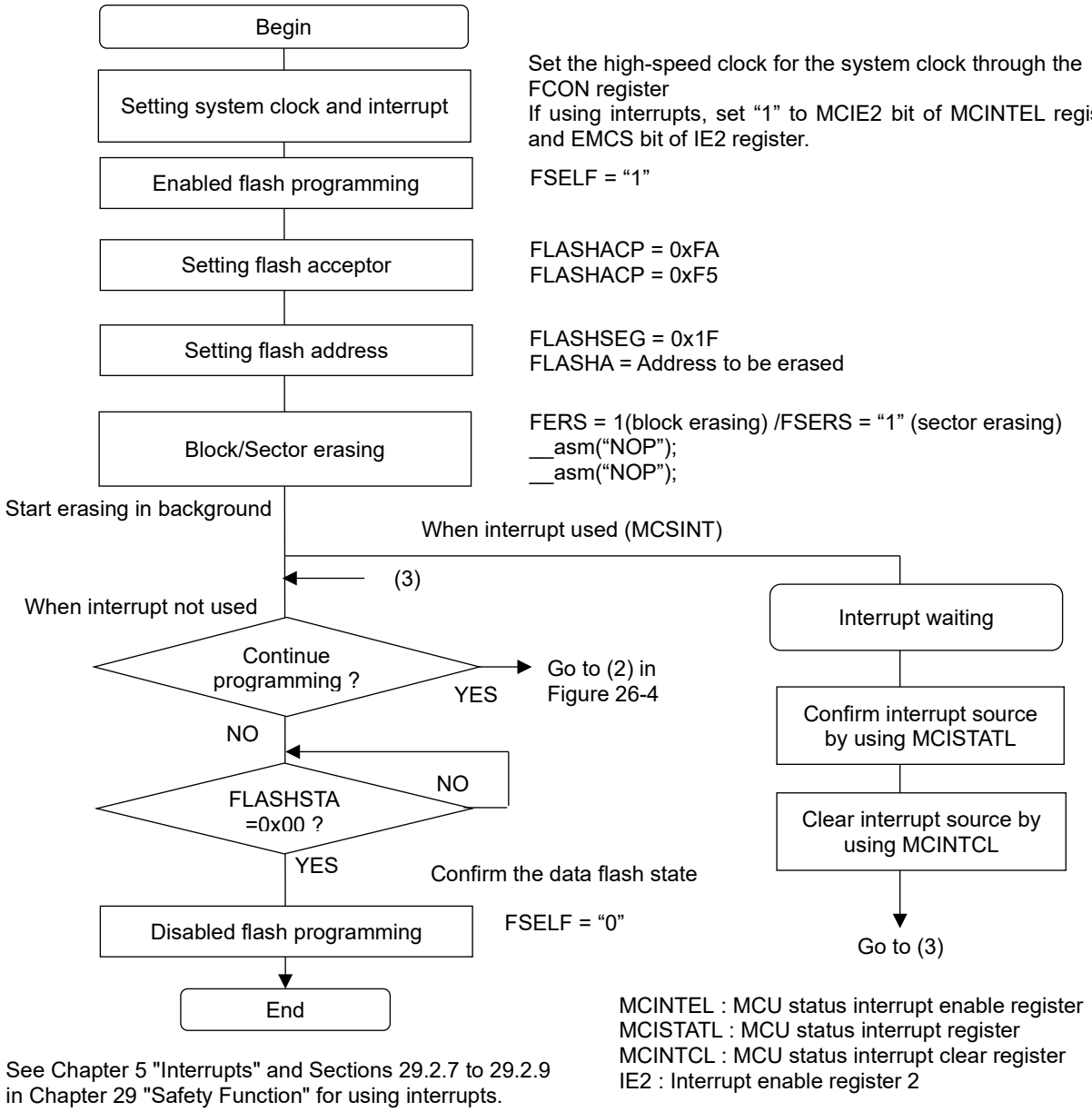


Figure 26-3 Flow Diagram for Erasing Data Flash Area

[Note]

- **The CPU continues program processing even while data flash erasing is in progress. An entering to the STOP mode is not available during the erasing. In addition, set FSELF bit of FLASHSLF register to "0" after the erasing is completed.**
- **The data flash area is unreadable during erasing.**
- **For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 26-4 shows the flow diagram for programming the data flash area.

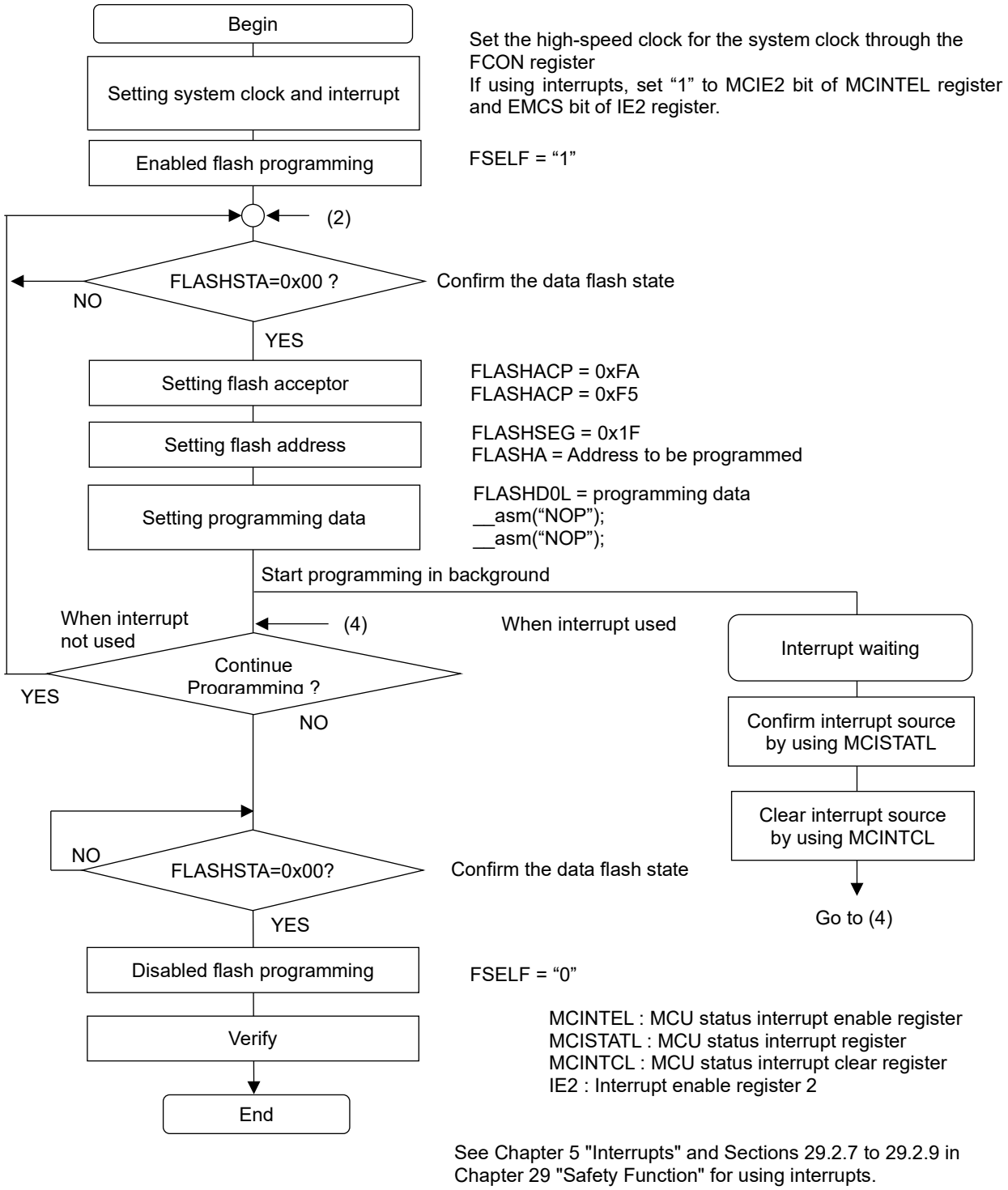


Figure 26-4 Flow Diagram for Programming Data Flash Area

[Note]

- **The CPU continues program processing even while data flash programming is in progress. An entering to the STOP mode is not available during the programming. In addition, set FSELF bit of FLASHSLF register to "0" (erase/program disabled) after the programming ended.**
- **The data flash area is unreadable during programming.**
- **For data programming setting, place two NOP instructions following the instruction used to set the programming data in FLASHD0L register.**

26.3.4 Notes on use of self-programming

Table 26-5 shows the notes on the use of self-programming (block erasing/sector erasing/programming).

Table 26-5 Notes on Use of Self-programming

Item	Notes
System clock during use of self-programming	Set to high-speed clock. See Chapter "6 Clock Generation Circuit" for enabling the high-speed clock oscillation and switching the system clock.
If power outage or forced termination due to a reset occurs	Data in flash memory is not guaranteed. Perform block/sector erase again then program data.
If LSI does not start up due to occurrence of power outage or forced termination during programming (*1)	Program the program again using on-chip debug emulator or ISP function.
Access to SFRs related flash control.	Do not perform to write to the FLASHACP/FLASHSLF register during self-programming; when FLASHSTA is not 0x0.

*1: While programming the block or sector including address 0:0000 of the program area is in progress.

26.4 In-System Programming Function

The In-System Programming (ISP) function is used to program a program memory space and data flash area through UART communication with an external device.

26.4.1 Programming Procedure

Figure 26-5 shows the flow diagram for programming the flash memory using the ISP function.

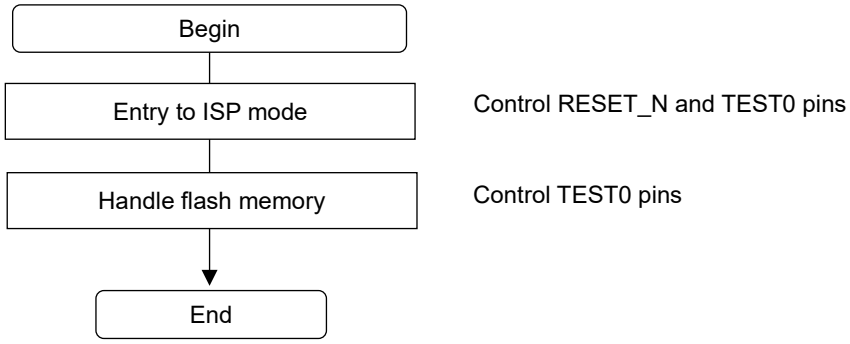


Figure 26-5 Flow Diagram for Programming Flash Memory Using ISP Function

26.4.2 Communication Method

Table 26-6 describes the communication method of the ISP function.

Table 26-6 ISP Function Communication Method

Item	Description
Pins for ISP function	RESET_N pin and TEST0 pin
Pins used when entering ISP mode	
Communication method	Half-duplex UART
Pin used for UART communication	TEST0 pin
UART Communication Format	8-bit length, LSB fast, 1 stop bit, no parity bit
Baud rate	Auto-detection between 4800bps to 2.0Mbps

26.4.3 Control Command

3-byte commands are used to make the communication in the ISP function.
Table 26-7 shows the ISP mode commands.

Table 26-7 ISP Mode Command List

Command		1st byte	2nd byte	3rd byte
Initial setting	Sending initial setting command (1)	0x1A	0x08	0x00
	Sending initial setting command (2)	0x1A	0x00	0x00
	Sending initial setting command (3)	0xC0	0x01	0x00
	Sending initial setting command (4)	0xC0	0x05	0x00
	Sending initial setting command (5)	0xC0	0x03	0x00
	Sending initial setting command (6)	0xCE	0x01	0x00
	Sending initial setting command (7)	0xCE	0x00	0x00
	Sending initial setting command (8)	0x96	0xFF	0xFF
	Sending initial setting command (9)	0x98	0xFF	0xFF
	Sending initial setting command (10)	0x9A	0xFF	0xFF
	Sending initial setting command (11)	0x9C	0xFF	0xFF
	Sending initial setting command (12)	0x9E	0xFF	0xFF
	Confirming command transmission completion(1)	0x01	(Read) 0xC0 or 0x80	(Read) 0x05
	Confirming command transmission completion(2)	0x91	(Read) 0x00	(Read) 0x00
Common setting	Segment setting command	0xC6	0x00~0x1F (segment value)	0x00
	Address setting command	0xC8	Lower 8 bits	Higher 8 bits
	BUSY confirmation command	0xC5	(Read) 0x1F	(Read) 0x01:BUSY 0x00:IDLE
Block erasing	Block erasing command	0xC2	0x05	0x00
Chip erasing	Chip erasing command	0xC2	0x06	0x00
For programing data	Data setting command H ; in program code area (higher 2bytes)	0xD2	Lower Byte	Higher Byte
	Data setting command L ; in program code area (lower 2bytes)	0xCA	Lower Byte	Higher Byte
	Data setting command D ; in data flash area	0xCA	1Byte data	0xFF
	Programming command	0xC2	0x04	0x00

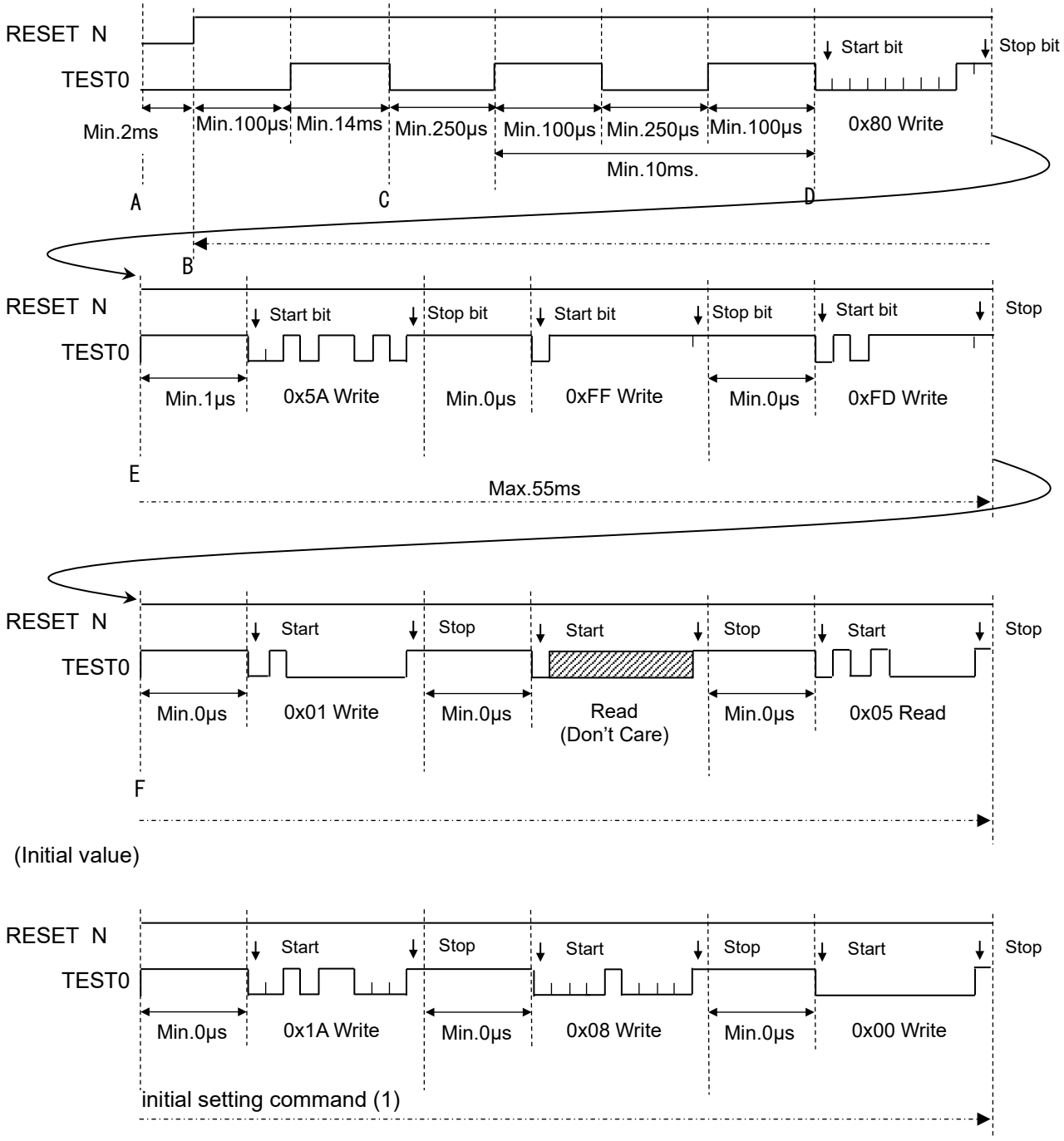
Command		1st byte	2nd byte	3rd byte
For verify	Expected data setting command H ; in program code area (higher 2bytes)	0xE4	Lower Byte	Higher Byte
	Expected data setting command L ; in program code area (lower 2bytes)	0xE2	Lower Byte	Higher Byte
	Expected data setting command D ; in data flash area	0xE4	1Byte data	0x00
	Verify command	0xC2	0x02	0x00
	Verify confirmation command ; collation result of expected data	0xE7	(Read) 0x03:OK 0x01:OK at current cycle, but has been NG in the past cycles. 0x02 or 0x00:NG	(Read) 0x00

[Note]

- **Accessing to the program code area is performed in units of four bytes. Set four byte boundaries (0H/4H/8H/CH) for lower four bits of the address. Accessing to the data flash area is performed in units of one byte.**

26.4.4 How to Entry ISP Mode

Figure 26-6 shows timing diagram to entry ISP mode.



A. System reset. Enter RESET_N="L" AND TEST0="L".
B. Raise the RESET_N from "L" to "H" during TEST0="L".
C. Transmit "L" level of 250 µs.min from TEST0 pin twice.
D. Transmit data 0x80 at the baud rate of UART communication set on the host side.
E. Transmit data in 0x5A→0xFF→0xFD order at the baud rate of UART communication set on the host side.
F. Transmit 0x01 at the baud rate of UART communication set on the host side, and if the third byte is 0x05, the transition to ISP mode is complete.
If the third byte is not 0x05, change the baud rate of the UART communication and repeat A~F.

Figure 26-6 Flow diagram to entry ISP mode

[Note]

- **Perform the process within 55ms from the ISP mode transition command (B in Figure 26-6) to the completion of the initial setting command transmission (1).**

26.4.5 Handling the Flash Memory

Figure 26-7 shows the flow diagram for erasing/programming the flash memory after transition to the ISP mode.

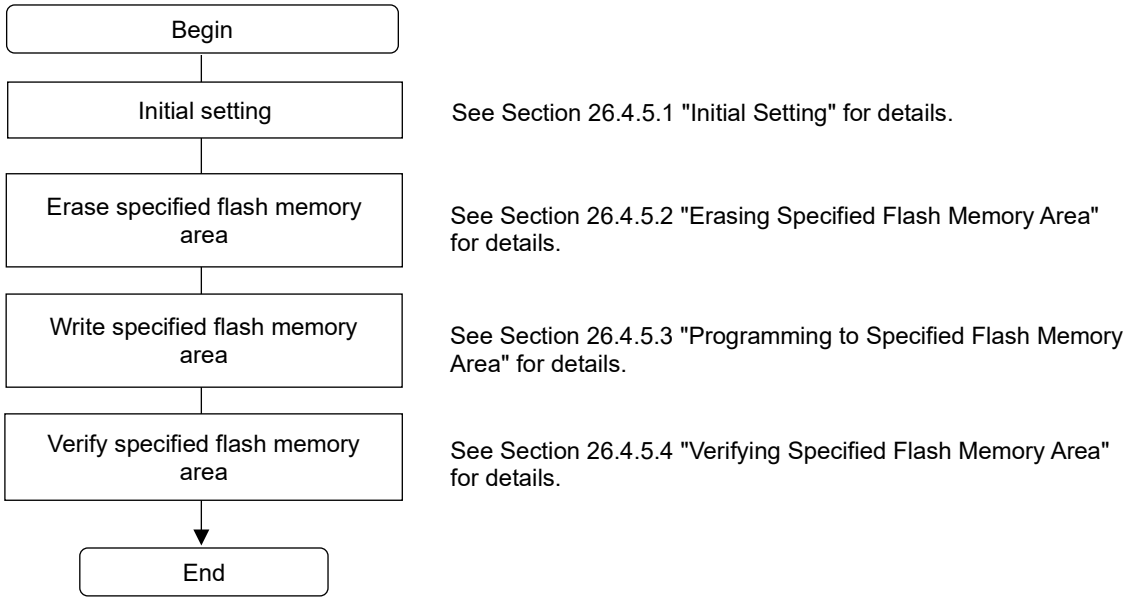


Figure 26-7 Flow Diagram for Erasing/Programming Flash Memory (Overview)

26.4.5.1 Initial Setting

Figure 26-8 shows the initial setting flow.

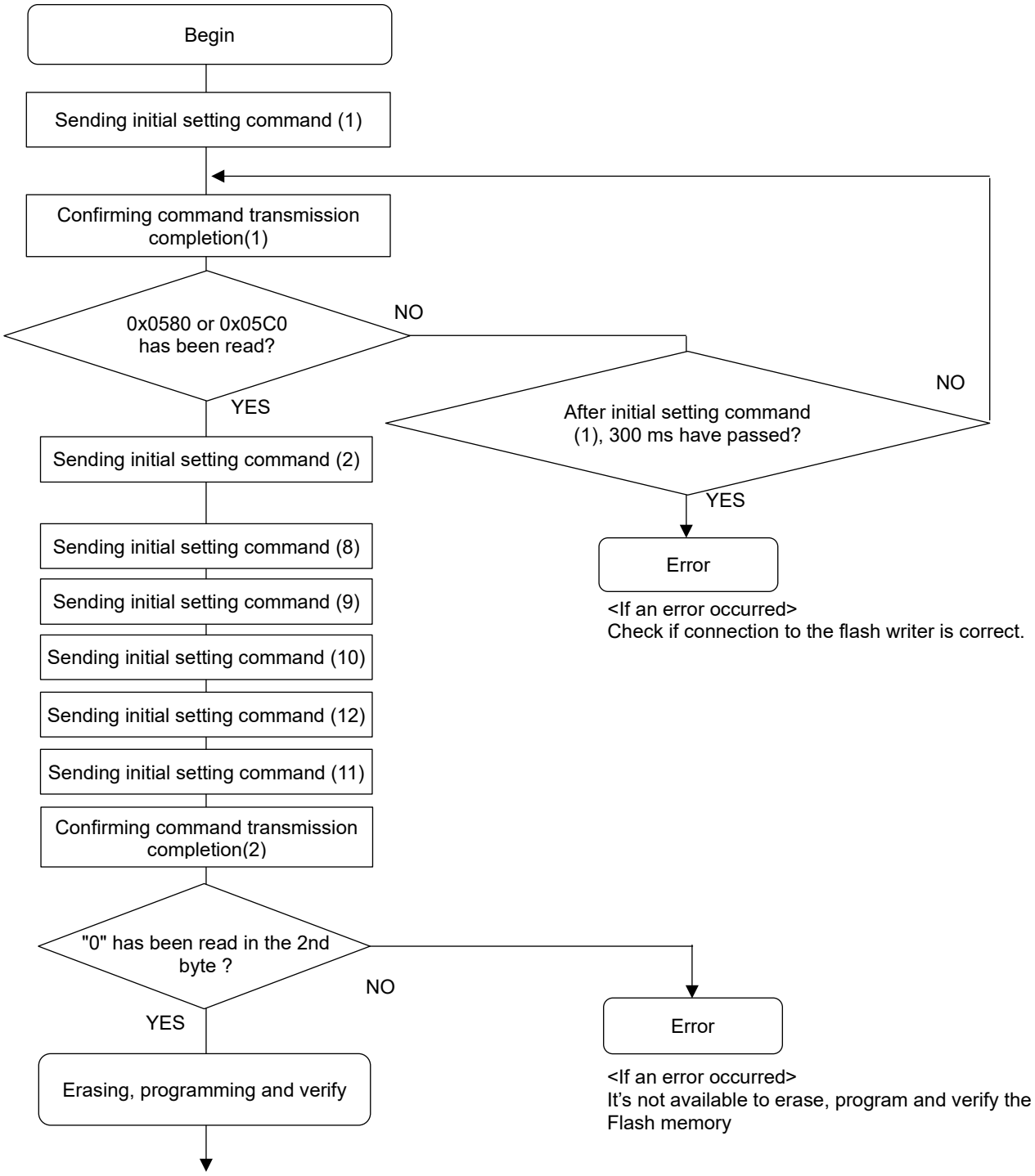


Figure 26-8 Initial Setting Flow Diagram

[Note]

- **Initial setting process should be processed within 1000 ms.**

26.4.5.2 Erasing Data in Specified Flash Memory Area

Figure 26-9 shows the flow diagram for erasing data in specified flash memory area.

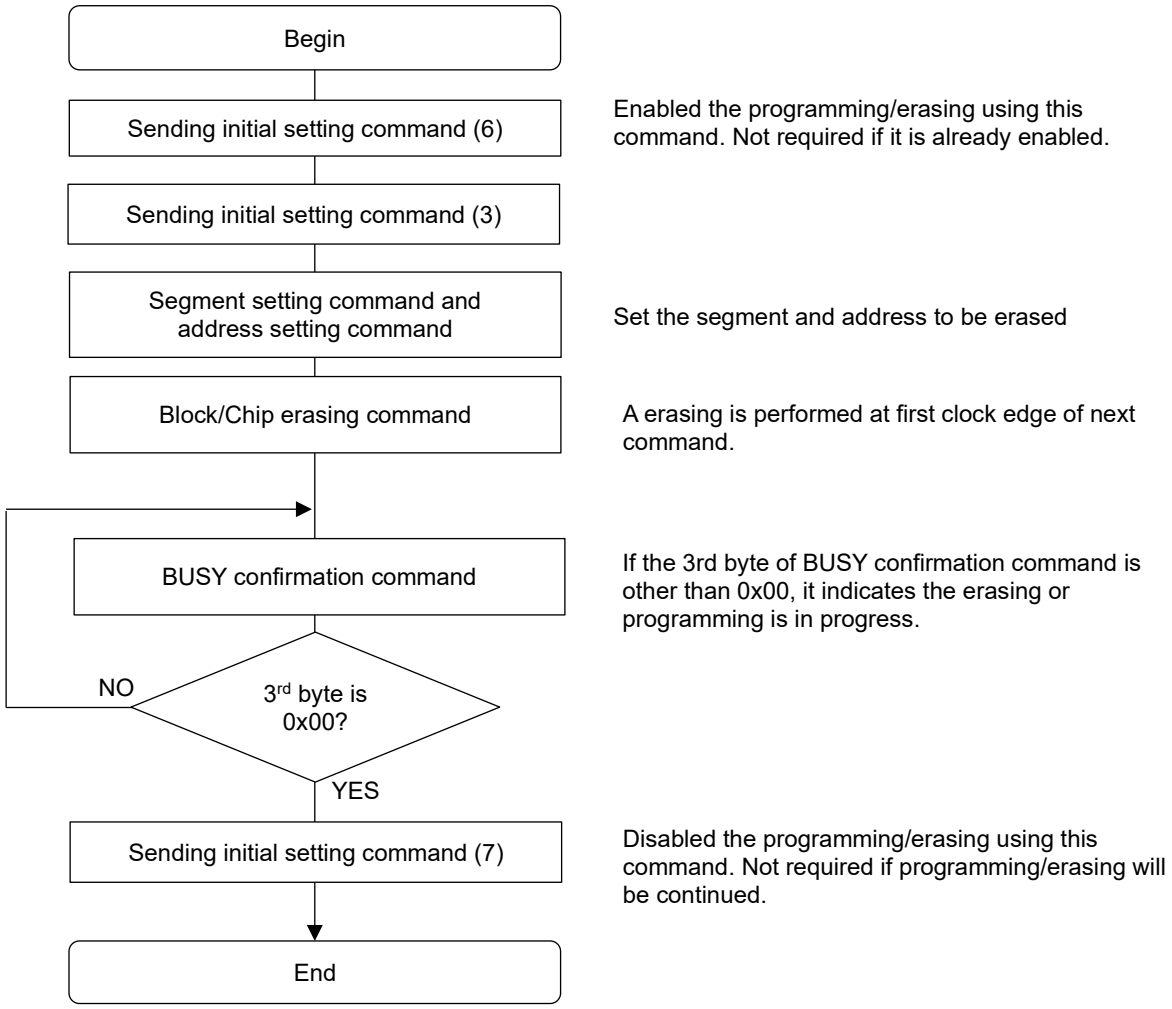


Figure 26-9 Flow Diagram for Erasing in Specified Flash Memory Area

[Note]

- **Erasure should be processed within 500ms.**

26.4.5.3 Programming to Specified Flash Memory Area

Figure 26-10 shows the flow diagram for programming to the specified flash memory area.

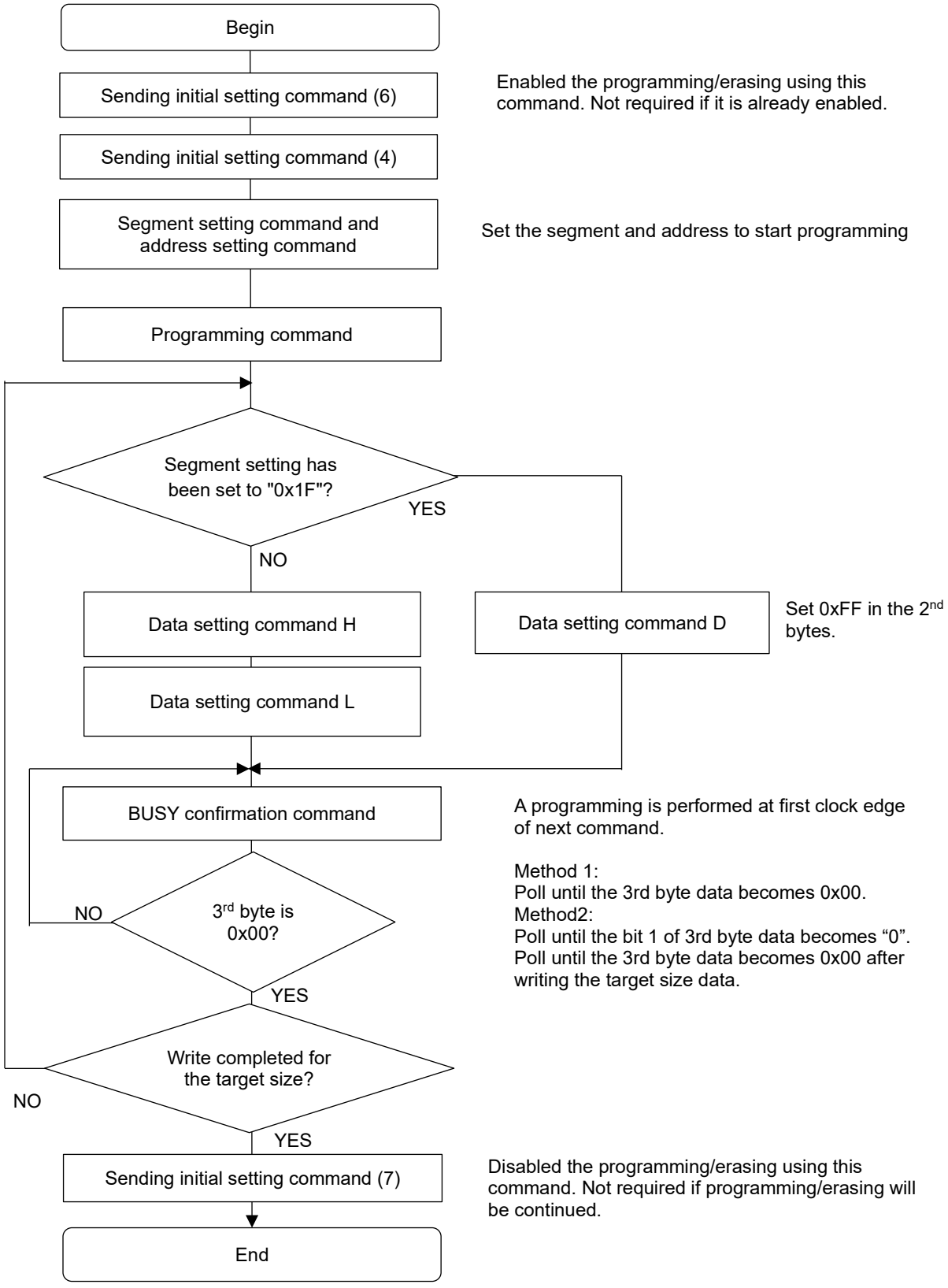


Figure 26-10 Flow Diagram for Programming to Specified Flash Memory Area

[Note]

- **Writing should be processed within 500ms. When writing multiple addresses, process from data setting to data setting of the next address or completion of sending initial setting command (7) within 500ms.**

26.4.5.4 Verifying data in Specified Flash Memory Area

Figure 26-11 shows the flow diagram for verifying data in the specified flash memory area.

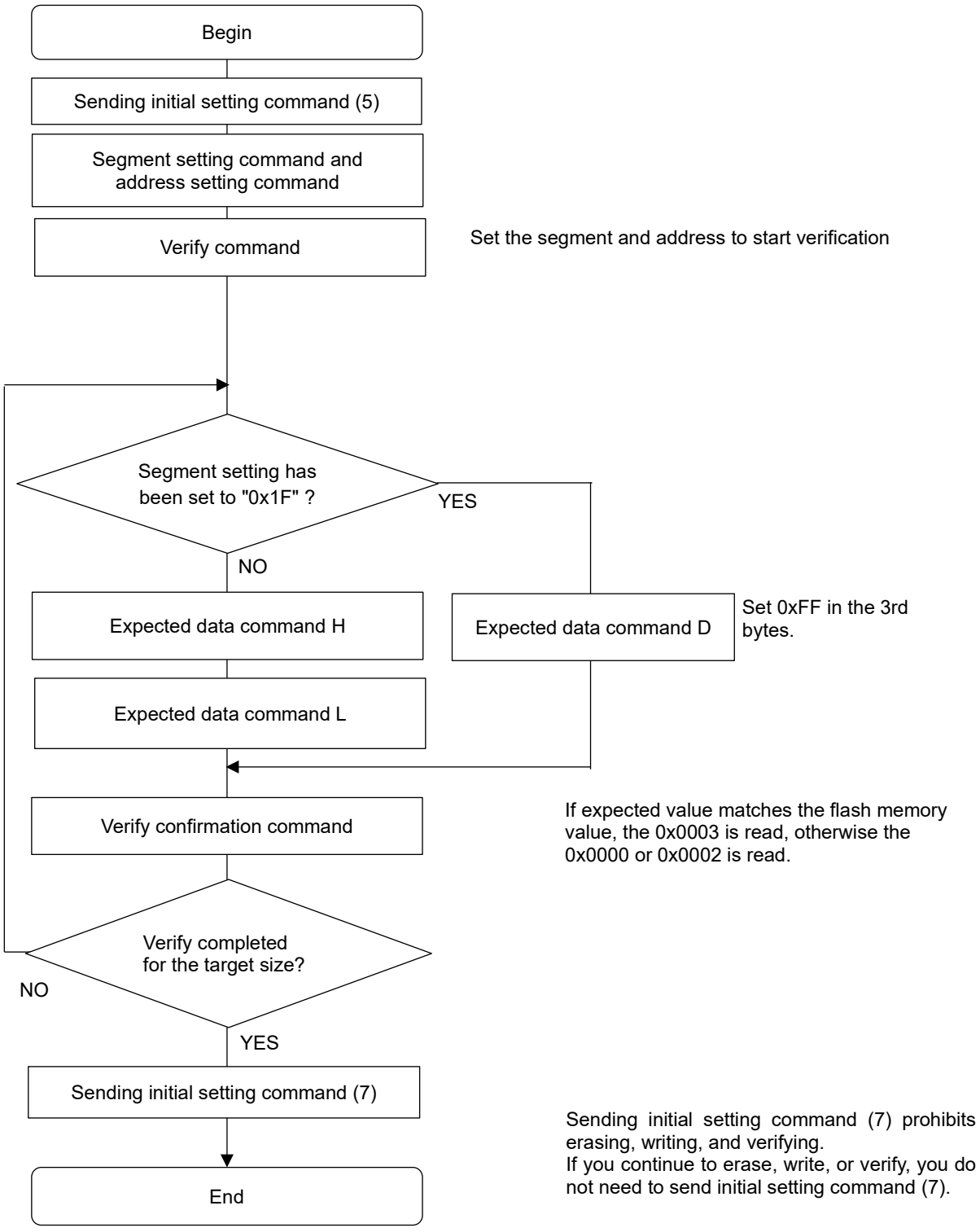


Figure 26-11 Flow Diagram for Verifying Specified Flash Memory Area

[Note]

- **Verify should be processed within 500ms. When verifying multiple addresses, process from data setting to data setting of the next address or completion of sending initial setting command (7) within 500ms.**

26.4.6 Advanced Control of Flash Memory Erasing/Programming

This section describes how to implement shorter Flash memory erasing/programing/verify time.

In Sections 26.4.5.2 and 26.4.5.3, after sending erase command or write data, the BUSY signal is confirmed and the next command is transferred considering the time until the BUSY signal is released (t_{busy}), it is possible to shorten the processing time by sending the following command.

26.4.6.1 Timing to transmit command of Advanced Control

When the LSI receives erasing or writing communication command, it issues an erasing/writing order to the flash memory. It requires the Busy time (t_{busy}) as an interval time to accept the next communication command. Therefore, transmit the communication commands for erasing/programming the Flash memory with an interval of longer than t_{busy}.

The timing of command transmit is calculated as follows.
Command transfer time : t_{cmd} = (10 [bit] / transfer rate[bps])
Wait time : t_{wait} = Busy time : t_{busy} – (t_{cmd} × number of commands)

(1) When the transfer rate is 1Mbps and programming data in program memory space:
Send the command so that the interval between receiving the " Writing data of program code (lower 2 bytes)" command is more than or equal to t_{busy}.

t_{cmd} = 10 bit / 1 Mbps = 10 [µs]
t_{wait} = t_{busy} – (t_{cmd} × 2) = 160 – 12 × 2 = 140 [µs]

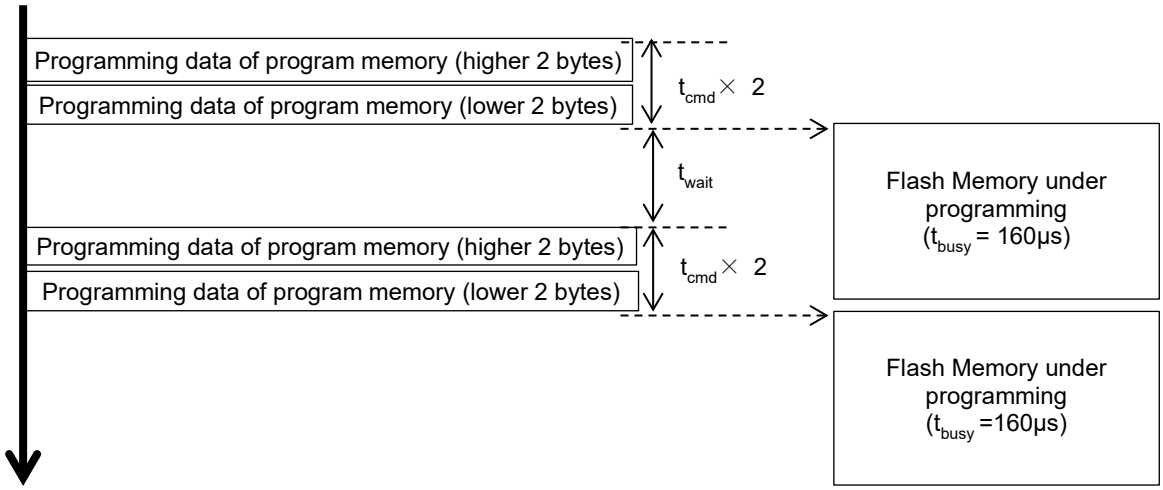


Figure 26-12 Advanced control #1 of programming the program code area

Figure 26-13 shows an example of using “data command H” instead of “BUSY confirmation command”.

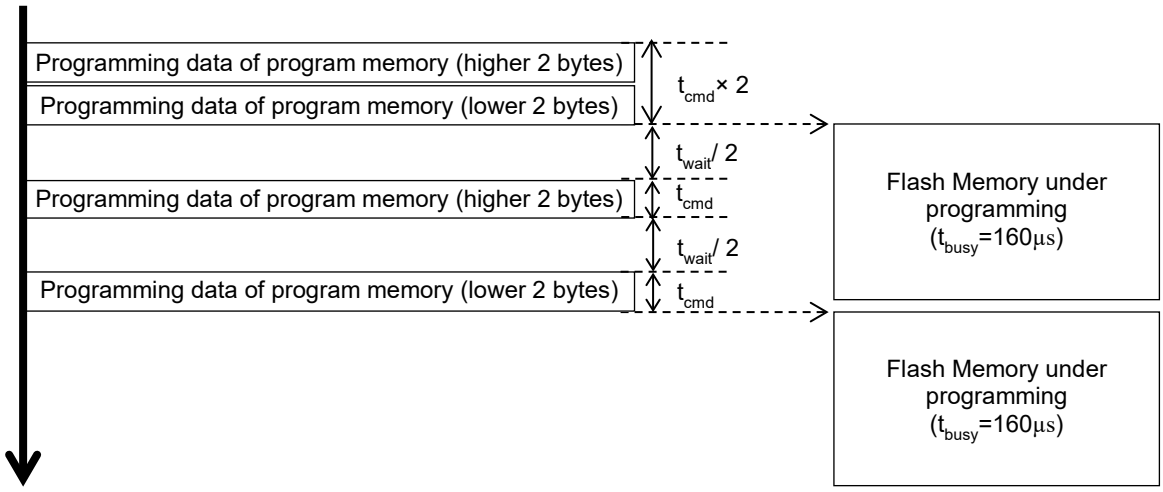


Figure 26-13 Advanced control #2 of programming the program code area

Figure 26-14 shows an example for programming to data flash area.
When the transfer rate is 1Mbps and programming data in data flash area:
Send the command “Data command D” so that the command acceptance interval is t_{busy} or more.

t_{cmd} = 10 bit / 1 Mbps = 10 [µs]
t_{wait} = t_{busy} – (t_{cmd} × 1) = 40 – 10 × 1 = 30 [µs]

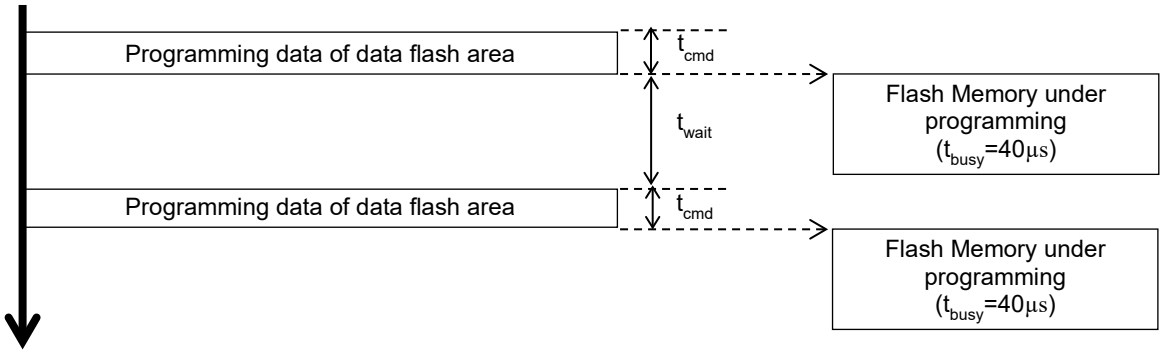


Figure 26-14 Advanced control of programming to the data flash

26.4.6.2 Timeout during advanced control

In ISP mode, there is a function to judge that it will time out after a certain period of time has elapsed and exit from ISP mode. After changing to ISP mode, issue one of the following commands at intervals of 2800 ms or less.

・Confirming command transmission completion(1)
・Confirming BUSY signal
・Confirming the result of matching the expected value

26.4.6.3 Erasing Data in Specified Flash Memory Area (Advanced control)

Figure 26-15 shows the flow diagram for erasing the specified flash memory area by the advanced control.

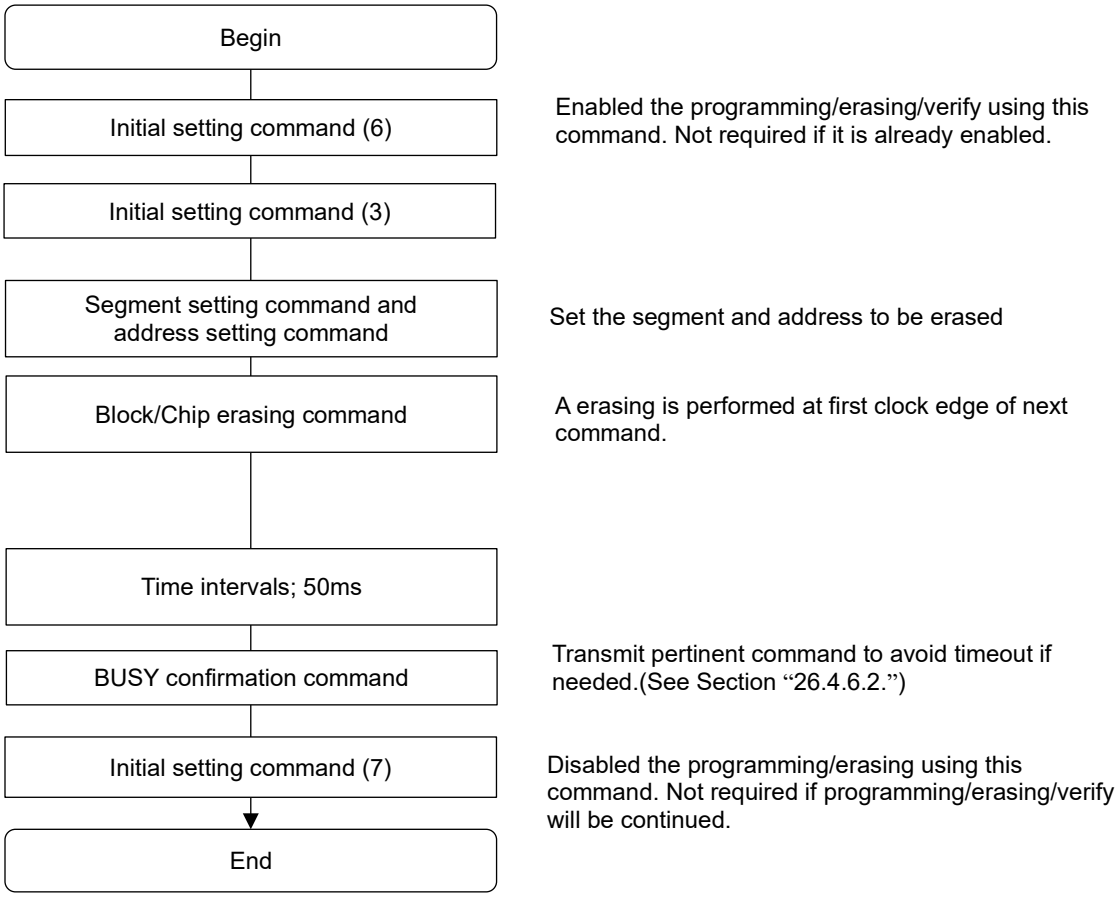


Figure 26-15 Flow Diagram for Erasing Specified Flash Memory Area (Advanced Control)

26.4.6.4 Programming to Specified Flash Memory Area (Advanced control)

Figure 26-16 shows the flow diagram for programming to the specified flash memory area by the advanced control.

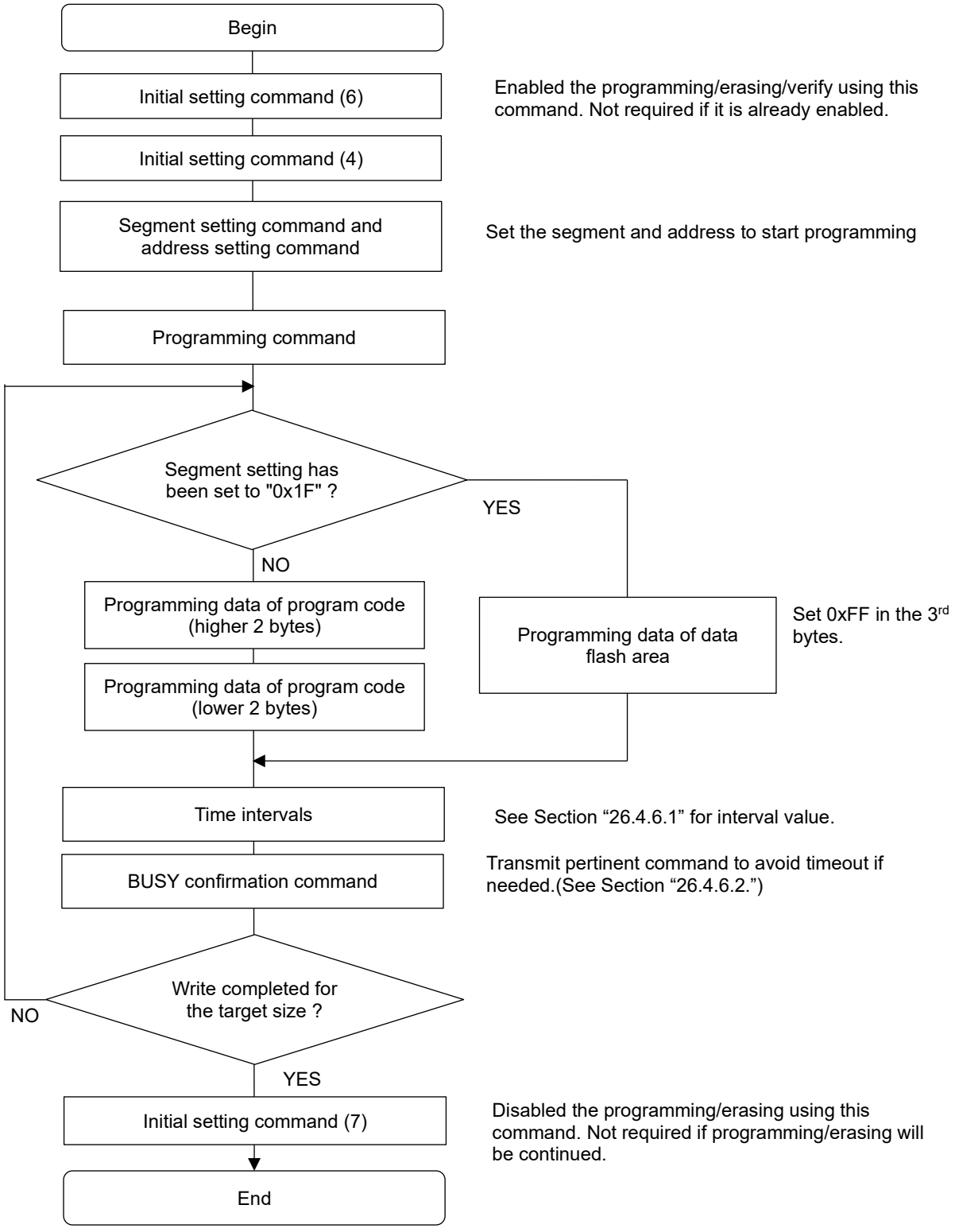


Figure 26-16 Flow Diagram for Programming Specified Flash Memory Area (Advanced Control)

Chapter 28 On-Chip Debug Function

28. On-Chip Debug Function

28.1 General Description

This function is used by connecting the host PC and LSI through the on-chip debug emulator (hereafter referred to as "On-chip emulator").
On-board debugging or programming is available by using the program development environment software (debugger) installed on the host PC.

28.1.1 Features

- The following debug functions are provided using the debugger by connecting LSI and On-chip emulator
  - Emulation
    - Real time emulation
    - Single step emulation
  - Trace
    - Branch trace (select Enable or Disable by code option)
  - Break
    - Hardware break point break (four points)
    - RAM data matching break
    - Sequential break
    - Trace overflow break(It is available if enabling branch trace function.)
    - Stack overflow/underflow break
    - Unused ROM area access break
    - RAM parity error break
  - Real time watch
  - CPU resource display/change
    - Program memory reference/disassembly
    - RAM and SFR display/change
    - Register display/change in the CPU
  - Program download
    - Program download/read/erase to/from flash memory
    - Data write/read/erase to/from data flash
  - Peripheral circuit operation continue/stop control
    Target peripheral circuits
    - External interrupt
    - 16-bit timer
    - Operational timer
    - UART
    - I^2C bus unit (master/slave)
    - Analog module (Analog comparator, Successive approximation type A/D converter, Low Level Detector)

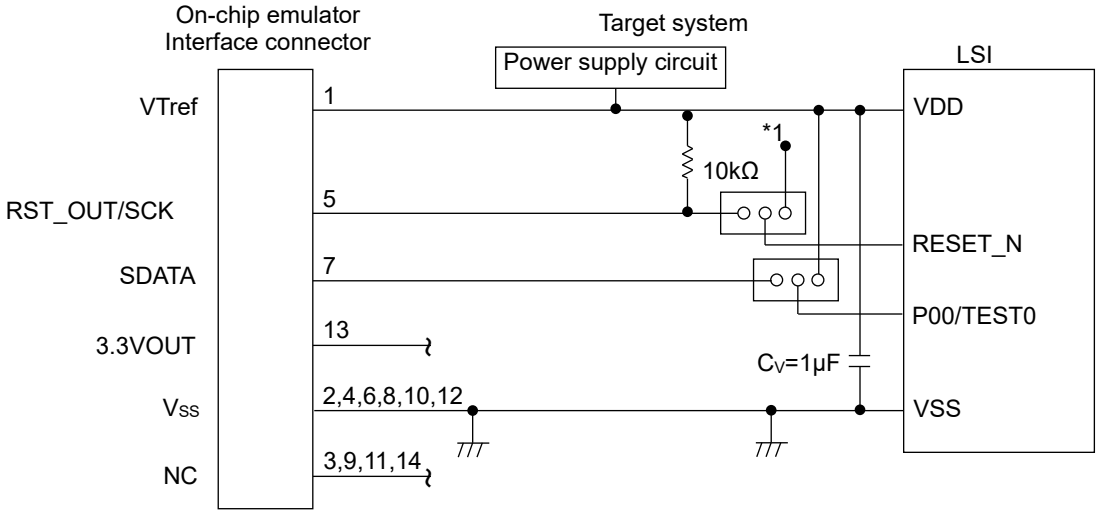
- The following program download function is provided using the flash multi-writer by connecting LSI and On-chip emulator.
  - Program download
    - Erasing/Programming the program memory space
    - Erasing/Programming the data flash memory area

28.1.2 Configuration

When using the on-chip debug function, use the power supply of the target system (VDD=4.5V~5.5V) to supply power to the LSI.

28.1.2.1 Using Power Supply of Target System (V_{DD}=4.5V to 5.5V)

Figure 28-1 shows a connection example when using the power supply (V_{DD}=4.5 V to 5.5 V) of On-chip emulator.



*1) Normal operation (reset IC, V_{DD}, etc.)

Figure 28-1 Connection Example

28.1.3 List of Pins

The following pins are used for the on-chip debug function.

Signal name	I/O	Function
RESET_N	I	On-chip debug function input signal
P00/TEST0	I/O	On-chip debug function input/output signal

28.2 How to Use On-chip Debug Function

See manual of the debugger for how to use the on-chip debug function using On-chip emulator and the debugger.
See manual of the flash multi-writer for how to download a program using On-chip emulator and flash multi-writer.

28.3 Precautions

[Note] on usage of the on-chip debug function.

- **Make RESET_N pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.**
- **Make P00/TEST0 pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.**
- **Do not program instruction codes into the LSI that set the P00/TEST0 pin to the output mode. If P00/TEST0 is set to the output mode before On-chip emulator performs read/write to/from the target chip, communication with On-chip emulator after that will be disabled. Also note that the input/output mode of P00/TEST0 is uninitialized by On-chip emulator.**
- **Validate the ROM code on user production board without the On-chip emulator.**
- **Disconnect On-chip emulator when measuring the current consumption of the target system. If On-chip emulator remains connected, the current consumption increases as the on-chip debug circuit inside the LSI works for the communication.**
- **LSI used to debug a program is not covered by the product warranty. Do not use the LSI for mass-production.**
- **A reset due to unused ROM area access does not occur in the on-chip debug mode regardless of code option settings.**
- **A RAM parity error reset does not occur in the on-chip debug mode and the break operation occurs instead.**
- **If the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including when the RAM is not initialized), a RAM parity error may occur even if the RAM area is not displayed.**
- **The all interrupts and watchdog timer operation always stop while the debugger is in the break state.**
- **If the LSI cannot be connected to the debugger, check that the LSI is supplied with a power supply (V_{DD}=4.5V~5.5V).**

28.4 Operation of Peripheral Circuits during breaks in the on-chip debug mode

The debugger allows users to choose whether to continue or stop operating the peripheral circuits during the break state on the debugger.
Table 28-1 shows the optional items, the target peripherals and how the operation is controlled.
Each optional item is displayed with a check box on the debugger. See manual of the debugger for more details on how to use the function.

Table 28-1 Peripheral controls during the break on the debugger

Optional item	Peripheral Circuit	Description
External Interrupt	External Interrupt	If the item is checked on, the target LSI accepts the external input during the break. If the item checked off, the target LSI does not accept the external input during the break.
General Timer	16-bit Timer	If the item is checked on, operation of the peripheral operation continues during the break. If the item checked off, operation of the peripheral stops during the break.
Operational Timer	Operational Timer	If the item is checked on, operation of the peripheral operation continues during the break. If the item checked off, operation of the peripheral stops during the break. Operational Timer can be controlled for each channels, refer to "9.2.25 OTM Common Break Control Registers" in "Chapter 9 Operational Timers".
I^2C Bus Unit (Master/Slave)	I^2C Bus Unit (Master/Slave)	If the item is checked on, operation of the peripheral operation continues during the break. If the item checked off, operation of the peripheral stops during the break.
Analog Module (CMP/ADC/LLD)	SA-ADC and Analog comparator and LLD	If the item is checked on, operation of the peripheral operation continues during the break. If the item checked off, operation of the peripheral stops during the break.

28.5 Reset in the On-Chip Debug Tool

By executing reset from the debug tool, RSTAT register POR bit is set to "1". However, LLD function is not reset. If it is necessary to start them from initial state, set these pertinent SFRs to initial value. Then execute reset from debug tools.

Chapter 29 Safety Function

29. Safety Function

29.1 General Description

The purpose of the safety function is to detect a fault by self-diagnosing the LSI and to stop the LSI safely.

29.1.1 Features

- Safety Functions on the LSI

Function Name	Description	Control by SFR
RAM guard	Protect from the miss-writing to the specified RAM area	Available
SFR guard	Protect from the miss-writing to the specified SFR	Available
Successive approximation type A/D converter test	Successive approximation type AD converter test function	Available
RAM parity error detection	RAM parity error check and generates a reset on error (enable/disable reset by SFR, with reset status flag and parity error flag)	Available
ROM unused area access reset	Make a reset in case the CPU executes an instruction in the unused area (enable/disable reset by the code option, with reset status flag)	-
Clock mutual monitoring	Monitor to check whether the oscillation of the high-speed and low-speed clocks are normal	Available
UART self-test function	Make the UART self-test	Available
I^2C self-test function	Make the I^2C self-test function	Available
WDT counter read	WDT counter read function	Available
Port output level self-test function	General port self-test function	Available
MCU status interrupt	Control interrupts generated by RAM parity error, automatic CRC calculation completion, and data flash erase/program completion.	Available

29.2 Description of Registers

29.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF0B0	RAM Guard Setting Register 0	RAMGD	-	R/W	8	0x00
0xF0B1 to 0xF0B3	Reserved	-	-	-	-	-
0xF0B4	SFR Guard Setting Register 0	SFRGD0L	SFRGD0	R/W	8/16	0x00
0xF0B5		SFRGD0H		R/W	8	0x00
0xF0B6	SFR Guard Setting Register 1	SFRGD1L	SFRGD1	R/W	8/16	0x00
0xF0B7		SFRGD1H		R/W	8	0x00
0xF0B8 to 0xFBB	Reserved	-	-	-	-	-
0xF0BC	RAM Parity Setting Register	RASFMOD	-	R/W	8	0x00
0xF0BD	Reserved	-	-	-	-	-
0xF0BE	Communication Test Setting Register 0	COMFT0L	COMFT0	R/W	8/16	0x00
0xF0BF		COMFT0H		R/W	8	0x00
0xF050	MCU Status Interrupt Enable Register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-	-	-	-	-
0xF052	MCU Status Interrupt Register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-	-	-	-	-
0xF054	MCU Status Interrupt Clear Register (L/H)	MCINTCLL	-	W	8	0x00
0xF055		MCINTCLH		W	8	0x00

29.2.2 RAM Guard Setting Register (RAMGD)

RAMGD is a SFR used to disable writing the RAM.
Data in the specified RAM area is protectable.

Address: 0xF0B0 (RAMGD)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								RAMGD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	RGD2	RGD1	RGD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2 to 0	RGD2 to RGD0	Select a protect area for writing on the RAM. 000: All RAM area writable and readable (Initial value) 001: 0x0:0EFC0 to 0x0:0EFFF (64 byte) is unwritable and readable 010: 0x0:0EF80 to 0x0:0EFFF (128 byte) is unwritable and readable 011: 0x0:0EF00 to 0x0:0EFFF (256 byte) is unwritable and readable 100: 0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable 101: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable) 110: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable) 111: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable)

29.2.3 SFR Guard Setting Register 0 (SFRGD0)

SFRGD0 is a SFR used to disable writing certain SFRs.
Data in the specified SFR area is protectable.

Address: 0xF0B4 (SFRGD0L/SFRGD0), 0xF0B5(SFRGD0H)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD0															
Byte	SFRGD0H								SFRGD0L							
Bit	-	-	-	-	-	-	-	-	-	-	SGD05	SGD04	SGD03	SGD02	SGD01	SGD00
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured writable/unwritable target SFRs
0: Target SFRs are writable and readable (Initial value)
1: Target SFRs are unwritable and readable

Bit No.	Bit symbol name	Description (target SFRs)
15 to 6	-	Reserved bits
5	SGD05	WDTMOD register; see Chapter 10 “Watchdog timer”
4	SGD04	BCKCONn and BRECONn registers (n=0 to 3); see Chapter 4 “Power management”
3	SGD03	RASFMOD register; see this chapter.
2	SGD02	SFRs described in Chapter 22 “LLD”
1	SGD01	SFRs described in Chapter 6 “Clock Generation Circuit”
0	SGD00	SFRs described in Chapter 5 “Interrupt”

29.2.4 SFR Guard Setting Register 1 (SFRGD1)

SFRGD1 is a SFR used to disable writing certain SFRs.
Data in the specified SFR area is protectable.

Address: 0xF0B6(SFRGD1L/SFRGD1), 0xF0B7(SFRGD1H)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD1															
Byte	SFRGD1H								SFRGD1L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	SGD12	SGD11	SGD10
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured writable/unwritable target SFRs in chapter 17. GPIO.
0: Target SFRs are writable and readable (Initial value)
1: Target SFRs are unwritable and readable

Bit No.	Bit symbol name	Description
15 to 3	-	Reserved bits
2	SGD12	SFRs related to the port 2
1	SGD11	SFRs related to the port 1
0	SGD10	SFRs related to the port 0

29.2.5 RAM Parity Setting Register (RASFMOD)

RASFMOD is a special function register (SFR) used to control the RAM parity error reset function.
The RAM parity error is detectable and the RAM parity error reset is generatable.
The reset flag by a RAM parity error can be checked by the reset status register (SRSTAT).
See Chapter 3 "Reset Function" for details about the reset flag.

Address: 0xF0BC(RASFMOD)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								RASFMOD							
Bit	-	-	-	-	-	-	-	-	PERF	-	-	-	-	-	-	PEREN
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	PERF	Check whether the RAM parity error occurs. Write "1" to this bit to clear. When PEREN is set to "1" to enable the parity error reset function, the reset status register (SRSTAT) can be used to check. 0: Disabled (Initial value) 1: Enabled
6 to 1	-	Reserved bits
0	PEREN	Enable or disable the RAM parity error reset function. 0: Disabled (Initial value) 1: Enabled

[Note]

- **If the RAM is not initialized when it is read, a parity error may occur. If you want to use parity error reset (check the parity error flag or enable parity error reset), initialize the entire RAM area before using it.**

29.2.6 Communication Test Setting Register (COMFT0)

COMFT0 is a SFR used to control the communication test function, which enables the loop back test with transmit data in the serial communication units. See Section 29.3.1 "Communication Function Self Test" for more details.
As the I^2C bus unit and the I^2C master are equipped with the function to read the transmit data, the function can be used for testing. For details, see Chapter 13 "I^2C Bus".

Address: 0xF0BE(COMFT0/COMFT0L), 0xF0BF(COMFT0H)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	COMFT0															
Byte	COMFT0H								COMFT0L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMFT01	CMFT00
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured enable/disable the self-test for target communication function.
0: Target function is disabled. (Initial value)
1: Target function is enabled.

Bit No.	Bit symbol name	Description (target)
15 to 2	-	Reserved bits
1	CMFT01	UART1
0	CMFT00	UART0

29.2.7 MCU Status Interrupt Enable Register (MCINTEL)

MCINTEL is a SFR used to control enabling/disabling three types of interrupt status on the microcontroller.

Address: 0xF050 (MCINTEL)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCINTEL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	MCI2E	-	MCI0E
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured enable/disable target interrupt.
0: Target interrupt is disabled. (Initial value)
1: Target interrupt is enabled.

Bit No.	Bit symbol name	Description (target interrupt)
7 to 3	-	Reserved bits
2	MCI2E	The interrupt at the completion of data flash erasing/programming.
1	-	Reserved bit
0	MCI0E	The interrupt at the occurrence of RAM parity error.

[Note]
- **If the RAM is not initialized when it is read, a parity error may occur.**

29.2.8 MCU Status Interrupt Register (MCISTATL)

MCISTATL is a read-only SFR used to indicate status of the three types of interrupts.
The MCI2S bit to MCI0S bit is initialized, in addition to reset function, by writing "1" to the same number of bit in the MCINTCL register.

Address: 0xF052 (MCISTATL)
Access: R
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCISTATL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	MCI2S	-	MCI0S
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is to indicate status of target interrupt.
0: Target interrupt has not been generated. (Initial value)
1: Target interrupt has been generated.

Bit No.	Bit symbol name	Description (target interrupt)
7 to 3	-	Reserved bits
2	MCI2S	The interrupt at the completion of data flash erasing/programming.
1	-	Reserved bit
0	MCI0S	The interrupt at the occurrence of RAM parity error.

[Note]

- **If the bit of each interrupt status is set to "1" and the same interrupt occurs again, the interrupt request will not be output. In order to output the interrupt request, write "1" to the same bit in MCINTCL register and clear the status bit to "0".**

29.2.9 MCU Status Interrupt Clear Register L/H (MCINTCLL, MCINTCLH)

MCINTCL is a write-only special function register (SFR) used to clear the MCU status interrupts.
If the MCI2C bit to MCI0C bit is set to "1", the interrupt request indicated by the same number of bit in the MCISTATL register gets cleared.
This register always returns "0x0000" for reading.

Address: 0xF054(MCINTCLL), 0xF055(MCINTCLH)
Access: W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	MCINTCLH								MCINTCLL							
Bit	MCIR	-	-	-	-	-	-	-	-	-	-	-	-	MCI2C	-	MCI0C
R/W	W	R	R	R	R	R	R	R	R	R	R	R	R	W	R	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits; bit 2 to 0 :
It is to indicate status of target interrupt.
Writing “0” : Invalid
Writing “1”: Target interrupt status gets cleared.

Bit No.	Bit symbol name	Description
15	MCIR	Request bit for the MCU status interrupt. Write "1" to this bit before returning from the interrupt routine. Writing "0": Invalid Writing "1": If an unhandled interrupt exists, it generates the interrupt request again.
14 to 3	-	Reserved bits
2	MCI2C	The interrupt at the completion of data flash erasing/programming.
1	-	Reserved bit
0	MCI0C	The interrupt at the occurrence of RAM parity error.

29.3 Description of Operation

29.3.1 Communication Function Self-Test

This self test is enabled by the COMFT0 register setting.
The communication function can be tested through the self test by internally connecting transmit and receive data of UART.
Before testing the communication, write "1" to the corresponding bit of the COMFT0 register.
Transmit side data output can be enabled/disabled by setting the mode (secondary to octonary function) of the general-purpose port.
For receive side data, it is not required to set the mode (2^{nd} to 8^{th} function) of the general-purpose port.
Figure 29-1 shows a concept diagram of the communication test. Figure 29-2 shows a flow chart of the communication test.

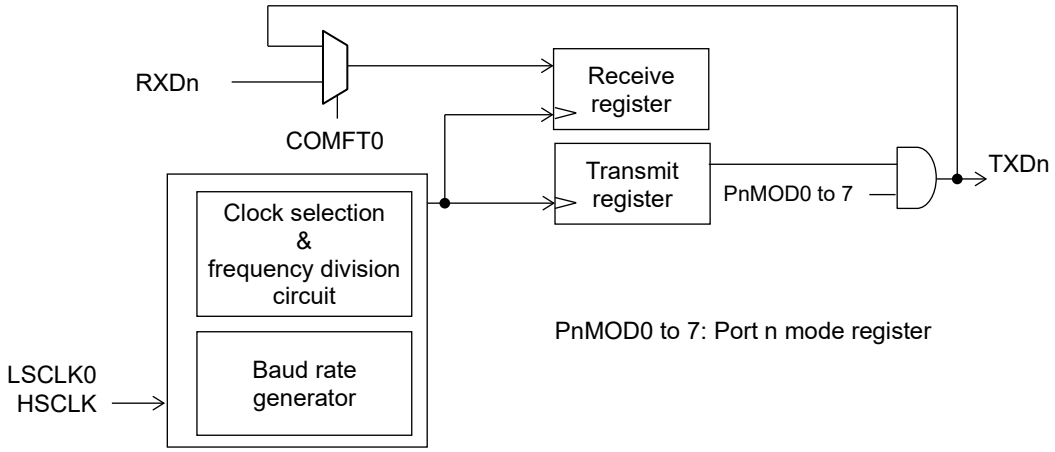


Figure 29-1 Communication Test Concept Diagram

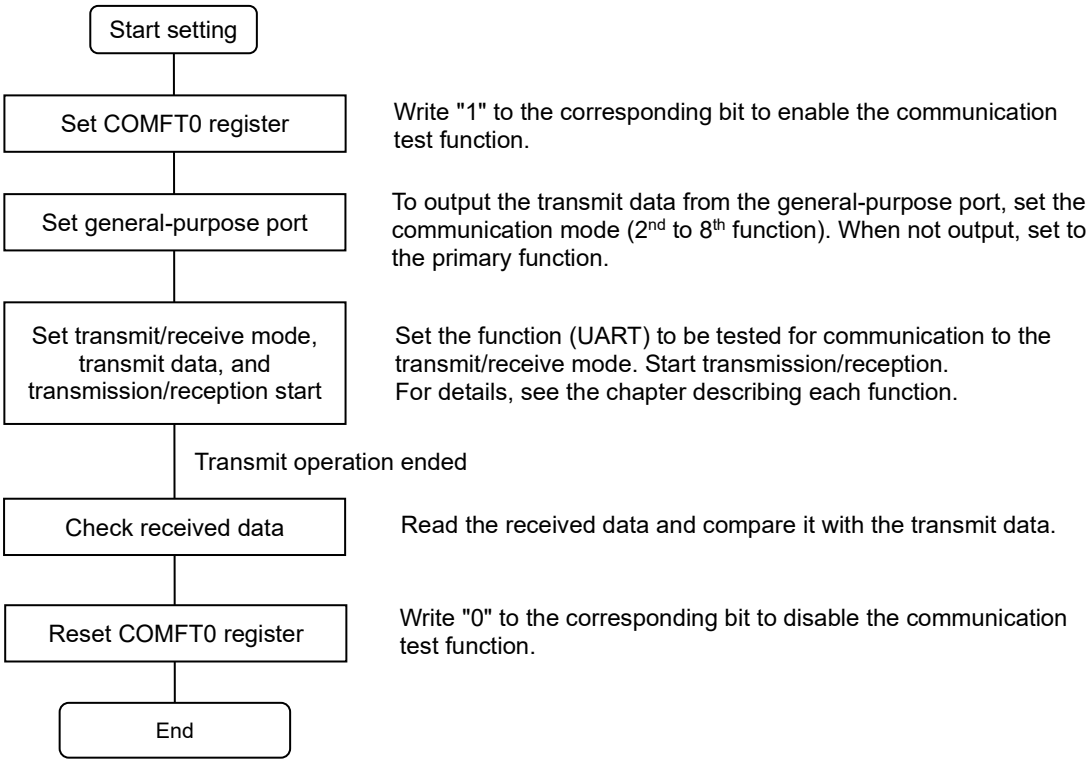


Figure 29-2 Communication Test Flow Chart

29.3.2 Unused ROM Area Access Reset Function

This function constantly monitors the program counter (PC) of the CPU. It generates the LSI reset when it detects that the program counter (PC) executes a program located outside of the area. This function can be enabled/disabled by the code option. The reset flag due to unused ROM area access can be confirmed with the SRSTAT register.
See Chapter 3 "Reset Function" for details of the reset flag.

<ROM unused area>

Program memory size : PC
32KB : 07FC0 to 0FFFF
16KB : 03FC0 to 0FFFF

29.3.3 Clock Mutual Monitoring Function

This function is used to monitor the low-speed clock (low-speed RC oscillation circuit) and high-speed clock (PLL oscillation circuit) to check if they are normally oscillating.
The 16-bit timer and operational timer are available to implement the function.
For details, see Section 9.3.10 "Clock Mutual Monitoring Function".

29.3.4 WDT Counter Read

The count value can be read from the watchdog timer counter register (WDTMC). Periodic checks of the count value allow confirmation that the watchdog timer is normally counting.
See Chapter 10 "Watchdog Timer" for its operation.

29.3.5 Port Output Level Test

When the general-purpose port is used as an output pin, the output data can be read by setting the input/output mode.
See Chapter 17 "General-purpose Port" for its operation.

29.3.6 Successive Approximation Type A/D Converter Test

The self test can be performed by A/D-converting the full scale, zero scale and internal reference voltage.
See Section 23.3.2 “Test function of Successive Approximation Type A/D Converter” for details.

Chapter 30 Code Option

30. Code Option

30.1 General Description

The code option is used to choose watchdog timer operation etc. depending on values written in the code option area of the program memory area.

The hardware automatically refers to data in the code option area when the microcontroller starts up due to one of system resets described below to set each function.
The code option area can be erased or programmed through the on-chip debug function, self-rewrite function of flash memory, or ISP function.

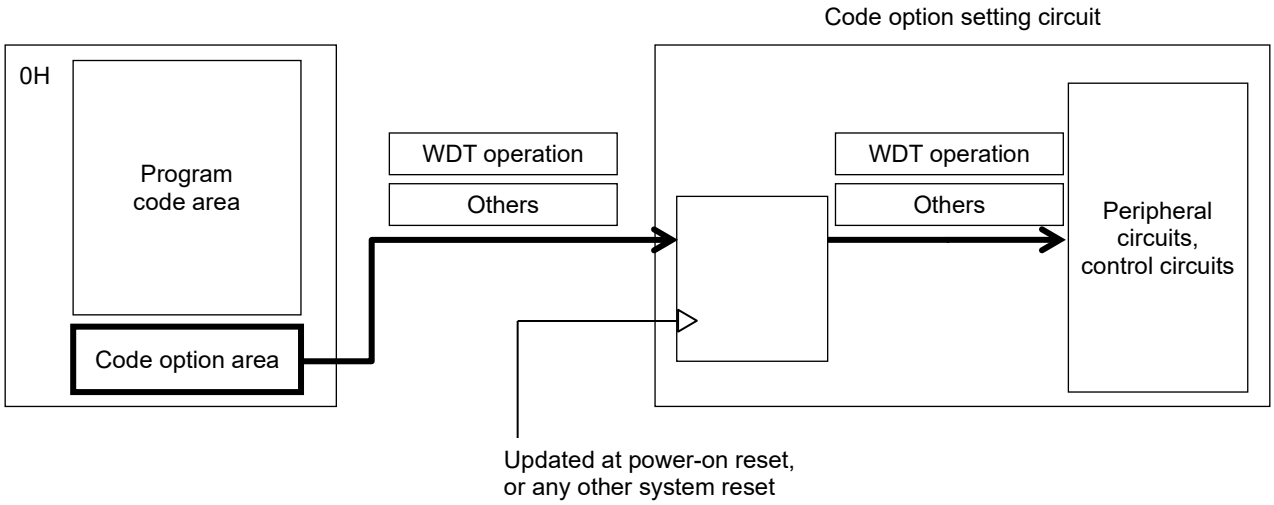


Figure 30-1 Code Option Overview

30.1.1 Function List

- Readable configured code options from SFRs
- Enabling or disabling the unused ROM area access reset
- Enabling or disabling the remapping function
- The software remap or hardware remap is selectable for the remap function
- Enabling or disabling the watchdog timer operation
- Enabling or disabling the watchdog timer operation at HALT

30.2 Description of Code Option

30.2.1 Reading from SFRs

The address of code option area is dependent of the size of the program memory space (flash memory).
The address of SFRs for reading is fixed.

SFR address	Register Name	Symbol name		R/W	Size
		Byte	Word		
0xF920	Code Option 0	CODEOP0L	CODEOP0	R	8/16
0xF921		CODEOP0H		R	8
0xF922	Code Option 1	CODEOP1L	CODEOP1	R	8/16
0xF923		CODEOP1H		R	8
0xF924	Code Option 2	CODEOP2L	CODEOP2	R	8/16
0xF925		CODEOP2H		R	8

[Note]

- **There are available to read the code option values from SFRs, if INITE flag bit of Reset Status Register (RSTAT) is "0".**

30.2.2 Code Option 0 (CODEOP0)

Address: (See Table 30-1)
SFR address for reading: 0xF920 (CODEOP0L/CODEOP0), 0xF921(CODEOP0H)
Access to SFR : R
Access size to SFR: 8/16 bit
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP0															
Byte	CODEOP0H								CODEOP0L							
Bit	-	-	-	PCERMD	-	-	-	REMAPMD	-	-	-	-	-	WDTPWMD0	-	WDTMD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits. Set "1" to all bits.
12	PCERMD	Select enable/disable the unused ROM area access reset. See Chapter 29.3.2 "Unused ROM Area Access Reset Function" for the unused ROM area access reset. 0: Disabled 1: Enabled (Initial value)
11 to 9	-	Reserved bits. Set "1" to all bits.
8	REMAPMD	Select to enable/disable the remapping function (software remap or hardware remap) operation. See Chapter 2.8 "Remapping Function" for details of the remapping function. 0: Enabled 1: Disabled (Initial value)
7 to 3	-	Reserved bits. Set "1" to all bits.
2	WDTPWMD0	Select to enable/disable the watchdog timer (WDT) operation in HALT/HALT-H mode, if WDTMD = “1”. 0: Disabled 1: Enabled (Initial value)
1	-	Reserved bit. Set "1" to this bit.
0	WDTMD	Select to enable/disable the watchdog timer (WDT) operation. 0: Disabled 1: Enabled (Initial value)

30.2.3 Code Option 1 (CODEOP1)

Address: (See Table 30-1)
SFR address for reading: 0xF922 (CODEOP1L/CODEOP1), 0xF923(CODEOP1H)
Access to SFR : R
Access size to SFR: 8/16 bit
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP1															
Byte	CODEOP1H								CODEOP1L							
Bit	-	-	-	-	-	-	-	-	TRAMUSE	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits. Set "1" to all bits.
7	TRAMUSE	Select whether to enable/disable the branch trace function among the debugging functions. When the branch trace function is enabled, 1 KB of the 2 KB of the installed RAM capacity is dedicated to the trace function during the debugger connection, so it cannot be used. (RAM usable range: 0EC00H~0EFFFH) If you disable the branch trace function, you can use the installed RAM capacity of 2 KB as it is. 0: Branch tracing function enabled 1: Branch tracing function disabled (initial value) To enable the branch trace function, refer to Chapter 30.4 "Branch Trace Function".
6 to 0	-	Reserved bits. Set "1" to all bits.

30.2.4 Code Option 2 (CODEOP2)

Address: (See Table 30-1)
SFR address for reading: 0xF924 (CODEOP2L/CODEOP2), 0xF925 (CODEOP2H)
Access to SFR : R
Access size to SFR: 8/16 bit
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP2															
Byte	CODEOP2H								CODEOP2L							
Bit	CREMAPMD	CRES2	CRES1	CRES0	CREA15	CREA14	CREA13	CREA12	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15	CREMAPMD	Control the initial value of Flash Remap Address Register (REMAPADD) at the system reset. 0: The initial value of the REMAPADD consists of CREA15 to 12 bits and CRES1 to 0 bits 1: The initial value of the REMAPADD is 0x00 If setting this bit to "0", The initial value of the REMAPADD consists of CREA15 to 12 bits and CRES1 to 0 bits. For details on REMAPADD, see Section 2.7.3 "Flash Remap Address Register (REMAPADD)". The MCU remaps to the address specified with the CREA15 to 12 bits and the CRES1 to 0 bits every time at the system reset. See also Section 2.8.3 "Code Option Remap". The remap function is enabled by setting REMAPMD bit of the Code Options 0.
14 to 12	CRES2 to CRES0	Set the initial values of RES2 to RES0 bits of the Flash Remap Address Register (REMAPADD). RES2 and RES1 are reserved bits. Set "0" to all bits.
11 to 8	CREA15 to CREA12	Set the initial values of REA15 to REA12 bits of the Flash Remap Address Register (REMAPADD).
7 to 0	-	Reserved bits. Set "1" to all bits.

CPU instruction execution start address after releasing the reset

Reset	REMAPMD	CREMAPMD	Remap function	CPU instruction execution start address
CPU reset (BRK instruction)	1	1	Disabled	0x0000
	1	0		
	0	1	Enabled Software remap	Address set in the REMAPADD register
	0	0		
System reset	1	1	Disabled	0x0000
	1	0		
	0	1	Enabled Code option remap	Initial data of the REMAPADD register (data set by the Code Options 2)
	0	0		

See Section 2.7.3 "Flash Remap Address Register (REMAPADD)" and Section 2.8.3 "Code Option Remap".

30.3 Code Option Data Setting

The address of code option area is dependent of the size of the program memory space (flash memory).
Table 30-1 shows addresses of code option areas for each product.

Table 30-1 List of Addresses of Code Option Areas for Each Product

Product name	Program memory space size	Code Option area	Address		
			CODEOP2	CODEOP1	CODEOP0
ML62Q2033/2043	16K byte	0x0:3FD0~0x0:3FFF	0x0:3FDC	0x0:3FDA	0x0:3FD8
ML62Q2035/2045	32K byte	0x0:7FD0~0x0:7FFF	0x0:7FDC	0x0:7FDA	0x0:7FD8

Figure 30-2 shows an example of a code option setting program (for products with the program memory space=32 Kbytes). The setting is described in the start-up file (ML6220xx.ASM) of each product.
Set every unused bit of the code option data area to "1".
For products with blank flash memory, every bit has been set to "1" as the factory default setting.

```
;-------------------------------------------------------------------
;           Setting the code-option data (for ML6220xx)
;-------------------------------------------------------------------
                cseg at 07fd0h          ; 32KB

                dw      0ffffh          ; 07fd0h
                dw      0ffffh          ; 07fd2h
                dw      0ffffh          ; 07fd4h
                dw      0ffffh          ; 07fd6h

                cseg #0 at 07fd8h
                dw      0eefeh          ; 07fd8h ;coop0
    ; Unused ROM area access reset disabled, remapping operation enabled, WDT operation disabled

                cseg #0 at 07fdah
                dw      0ffffh          ; 07fdah ;coop1
    ; Disabled debugger branch tracing

                cseg #0 at 07fdch
                dw      00dffh          ; 07fdch ;coop2
    ; Setting the remap address during a system reset (0:d000)

                cseg #0 at 07fdeh
                dw      0ffffh          ; 07fdeh ;
                dw      0ffffh          ; 07fe0h ;
                dw      0ffffh          ; 07fe2h ;
                dw      0ffffh          ; 07fe4h ;
                dw      0ffffh          ; 07fe6h ;
                dw      0ffffh          ; 07fe8h ;
                dw      0ffffh          ; 07feah ;
                dw      0ffffh          ; 07fech ;
                dw      0ffffh          ; 07feeh ;
                dw      0ffffh          ; 07ff0h ;
                dw      0ffffh          ; 07ff2h ;
                dw      0ffffh          ; 07ff4h ;
                dw      0ffffh          ; 07ff6h ;
                dw      0ffffh          ; 07ff8h ;
                dw      0ffffh          ; 07ffah ;
                dw      0ffffh          ; 07ffch ;
                dw      0ffffh          ; 07ffeh ;
```

Figure 30-2 Example of Code Option Data Program (for Products with the Program Memory Space = 32 Kbytes)

[Note]

- **For the code option data definition, always use the dw directive instruction to configure the data in the unit of word.**

30.4 Branch Trace Function

If you set the CODEOP1 TRAMUSE bit to 0, the branch trace function is enabled when a debugger is attached.
There are the following differences in this LSI depending on whether the branch trace function is enabled or disabled.

・Product ID

When the branch trace function is enabled, the extension bit 0(EX0) of the product ID becomes "1".
In addition, the LSI target name is appended with "R" after the product name.

・RAM area of data memory

Since the branch trace function occupies 1 KB of the 2 KB RAM area (0E800H~0EFFFH) of the data memory space, the RAM area that can be used as data memory is 1 KB (0EC00H~0EFFFH).

Since the RAM area that can be used as data memory differs when the branch trace function is enabled, use the LSI target name appended "R" to the end of the product name during compilation.

[Note]

- **At the time of shipment or when the program is erased, the code options are also initialized and the branch trace function is disabled.**
  If you want to enable branch trace function for an LSI that is in disabled branch trace function state, connect to a debugger with the normal product name and enable branch trace function.

Chapter 31 Auxiliary Function

31. Auxiliary Function

31.1 General Description

- Indication of Product ID
- Indication of unique chip ID (32bit); reading from FLASH

31.2 Description of Registers

31.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF930	Product ID register 0	PID0L	PID0	R	8/16	*1
0xF931		PID0H		R	8	*1
0xF932	Product ID register 1	PID1L	PID1	R	8/16	0x22
0xF933		PID1H		R	8	0x06

*1: It depends on product.

31.2.2 Product ID Register 0,1 (PID0, PID1)

This is a SFR to indicate product ID.

Address : 0xF930 (PID0/PID0L), 0xF931(PID0H), 0xF932 (PID1/PID1L), 0xF933(PID1H)
Access : R
Access size : 8/16 bit
Initial value : PID1: 0622, PID0:0xxx

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PID1															
Byte	PID1H								PID1L							
Bit	-	-	-	-	d53	d52	d51	d50	d43	d42	d41	d40	d33	d32	d31	d30
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PID0															
Byte	PID0H								PID0L							
Bit	d23	d22	d21	d20	d13	d12	d11	d10	d03	d02	d01	d00	ex3	ex2	ex1	ex0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0	0	0	0/1

Bit No.	Bit symbol name	Description
15 to 4	-	It indicates 6-digit number of product name. The upper 20-bit number is fixed at “0x6220”. The lower 8-bit number depends on the product name.
3 to 0	-	It indicates 1 character from A to F as extended identifier of product. If the product name has “P” or “T”, it is ignored. If branch trace is enabled (TRAMUSE bit = 0) in the code option, ex0 becomes "1".

Ex) ML62Q2045 : “0x0622_0450”

31.3 Description of Operation

31.3.1 How to Confirm Unique ID

This LSI chip has unique ID with 32-bit. It is to read from test area in data memory space.
Its address is different for each product.

Table 31-1 Address to read for each product

Product name	Program Memory size	Data Memory Space Address to read.
ML62Q2033/2043	16KByte	0x0:43F7 to 4
ML62Q2035/2045	32KByte	0x0:83F7 to 4

Appendix

Appendix A Resister List

The SFR list is show below. Access "Reserved" register is not guaranteed. Please do not access them.
Initial value with *1 depend on code option that is set. See Chapter "30 Code option".
Initial value with *2 depend on product. See Chapter "31 Auxiliary Function".

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF000	Data segment register	DSR	-	R/W	8	0x00
0xF001	Reserved	-		-	-	-
0xF002	High-speed clock mode register	FHCKMODL	FHCKMOD	R/W	8/16	0x00
0xF003		FHCKMODH		R/W	8	0x03
0xF004	Reserved	-	-	-	-	-
0xF005	Reserved	-		-	-	-
0xF006	Clock control register	FCON	FCONW	R/W	8/16	0x00
0xF007		FCON1		R/W	8	0x00
0xF008	High-speed clock wake up time setting register	FHWUPT	-	R/W	8	0x00
0xF009	Reserved	-		-	-	-
0xF00A to 0xF00F	Reserved	-	-	-	-	-
0xF010	Watchdog timer control register	WDTCON	-	R/W	8	0x00
0xF011	Reserved	-		-	-	-
0xF012	Watchdog timer mode register	WDTMOD	-	R/W	8	0x06
0xF013	Reserved	-		-	-	-
0xF014	Watchdog timer counter register	WDTMCL	WDTMC	R	8/16	0x00
0xF015		WDTMCH		R	8	0x00
0xF016	Watchdog timer status register	WDTSTA	-	R	8	0x01
0xF017	Reserved	-		-	-	-
0xF018	Stop code acceptor	STPACP	-	W	8	0x00
0xF019	Reserved	-		-	-	-
0xF01A	Standby control register	SBYCONL	SBYCON	W	8/16	0x00
0xF01B		-		-	-	-
0xF01C	Standby prohibition flag register	SBYEFLG	-	R	8	0x00
0xF01D	Reserved	-		-	-	-
0xF01E	Reserved	-	-	-	-	-
0xF01F	Reserved	-		-	-	-
0xF020	Interrupt enable register 01	IE0	IE01	R/W	8/16	0x00
0xF021		IE1		R/W	8	0x00
0xF022	Interrupt enable register 23	IE2	IE23	R/W	8/16	0x00
0xF023		IE3		R/W	8	0x00
0xF024	Interrupt enable register 45	IE4	IE45	R/W	8/16	0x00
0xF025		IE5		R/W	8	0x00
0xF026	Interrupt enable register 67	IE6	IE67	R/W	8/16	0x00
0xF027		IE7		R/W	8	0x00
0xF028	Interrupt request register 01	IRQ0	IRQ01	R/W	8/16	0x00
0xF029		IRQ1		R/W	8	0x00
0xF02A	Interrupt request register 23	IRQ2	IRQ23	R/W	8/16	0x00
0xF02B		IRQ3		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF02C	Interrupt request register 45	IRQ4	IRQ45	R/W	8/16	0x00
0xF02D		IRQ5		R/W	8	0x00
0xF02E	Interrupt request register 67	IRQ6	IRQ67	R/W	8/16	0x00
0xF02F		IRQ7		R/W	8	0x00
0xF030	Interrupt level control enable register	ILEN	-	R/W	8	0x00
0xF031	Reserved	-		-	-	-
0xF032	Current interrupt level management register	CIL	-	R/W	8	0x00
0xF033	Interrupt level mask register	MCIL		R/W	8	0x00
0xF034	Interrupt level control register 0	ILC00	ILC0	R/W	8/16	0x00
0xF035		ILC01		R/W	8	0x00
0xF036	Interrupt level control register 1	ILC10	ILC1	R/W	8/16	0x00
0xF037		ILC11		R/W	8	0x00
0xF038	Interrupt level control register 2	ILC20	ILC2	R/W	8/16	0x00
0xF039		ILC21		R/W	8	0x00
0xF03A	Interrupt level control register 3	ILC30	ILC3	R/W	8/16	0x00
0xF03B		ILC31		R/W	8	0x00
0xF03C	Interrupt level control register 4	ILC40	ILC4	R/W	8/16	0x00
0xF03D		ILC41		R/W	8	0x00
0xF03E	Interrupt level control register 5	ILC50	ILC5	R/W	8/16	0x00
0xF03F		ILC51		R/W	8	0x00
0xF040	Interrupt level control register 6	ILC60	ILC6	R/W	8/16	0x00
0xF041		ILC61		R/W	8	0x00
0xF042	Interrupt level control register 7	ILC70	ILC7	R/W	8/16	0x00
0xF043		ILC71		R/W	8	0x00
0xF044	External interrupt control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	-	-	-	-	-
0xF047	Reserved	-		-	-	-
0xF048	External interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A	Reserved	-	-	-	-	-
0xF04B	Reserved	-		-	-	-
0xF04C	External interrupt port selection register 0	EIPSEL0L	EIPSEL0	R/W	8/16	0x00
0xF04D		EIPSEL0H		R/W	8	0x00
0xF04E	Reserved	-	-	-	-	-
0xF04F	Reserved	-		-	-	-
0xF050	MCU Status Interrupt Enable Register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-		-	-	-
0xF052	MCU Status Interrupt Register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-		-	-	-
0xF054	MCU Status Interrupt Clear Register (L/H)	MCINTCLL	-	W	8	0x00
0xF055		MCINTCLH		W	8	0x00
0xF056	Reserved	-	-	-	-	-
0xF057	Reserved	-		-	-	-
0xF058	Reset status register	RSTATL	RSTAT	R/W	8/16	Undefined
0xF059		RSTATH		R/W	8	Undefined

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF05A	Safety function reset status register	SRSTAT	-	R/W	8	Undefined
0xF05B	Reserved	-		-	-	-
0xF05C	Software reset acceptor	SOFTRACP	-	W	8	0x00
0xF05D	Reserved	-		-	-	-
0xF05E	Software reset control register	SOFTRCON	-	R/W	8	0x00
0xF05F	Reserved	-		-	-	-
0xF060 to 0xF06F	Reserved	-	-	-	-	-
0xF070	Block clock control register 0	BCKCON0L	BCKCON0	R/W	8/16	0x00
0xF071		BCKCON0H		R/W	8	0x00
0xF072	Block clock control register 1	BCKCON1L	BCKCON1	R/W	8/16	0x00
0xF073		BCKCON1H		R/W	8	0x00
0xF074	Block clock control register 2	BCKCON2L	BCKCON2	R/W	8/16	0x00
0xF075		BCKCON2H		R/W	8	0x00
0xF076	Block clock control register 3	BCKCON3L	BCKCON3	R/W	8/16	0x00
0xF077		BCKCON3H		R/W	8	0x00
0xF078	Block reset control register 0	BRECON0L	BRECON0	R/W	8/16	0x00
0xF079		BRECON0H		R/W	8	0x00
0xF07A	Block reset control register 1	BRECON1L	BRECON1	R/W	8/16	0x00
0xF07B		BRECON1H		R/W	8	0x00
0xF07C	Block reset control register 2	BRECON2L	BRECON2	R/W	8/16	0x00
0xF07D		BRECON2H		R/W	8	0x00
0xF07E	Block reset control register 3	BRECON3L	BRECON3	R/W	8/16	0x00
0xF07F		BRECON3H		R/W	8	0x00
0xF080 to 0xF08F	Reserved	-	-	-	-	-
0xF090	Flash address register	FLASHAL	FLASHA	R/W	8/16	0xFF
0xF091		FLASHAH		R/W	8	0xFF
0xF092	Flash data register 0	FLASHD0L	FLASHD0	R/W	8/16	0xFF
0xF093		FLASHD0H		R/W	8	0xFF
0xF094	Reserved	-	-	-	-	-
0xF095	Reserved	-		-	-	-
0xF096	Flash control register	FLASHCON	-	W	8	0x00
0xF097	Reserved	-		-	-	-
0xF098	Flash acceptor	FLASHACP	-	W	8	0x00
0xF099	Reserved	-		-	-	-
0xF09A	Flash segment register	FLASHSEG	-	R/W	8	0x10
0xF09B	Reserved	-		-	-	-
0xF09C	Flash self register	FLASHSLF	-	R/W	8	0x00
0xF09D	Reserved	-		-	-	-
0xF09E	Flash status register	FLASHSTA	-	R	8	0x00
0xF09F	Reserved	-		-	-	-
0xF0A0	Flash remap address register	REMAPADD	-	R/W	8	*1
0xF0A1	Reserved	-		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF0A2 to 0xF0AF	Reserved	-	-	R/W	8	0x00
0xF0B0	RAM Guard Setting Register 0	RAMGD	-	R/W	8	0x00
0xF0B1	Reserved	-		-	-	-
0xF0B2	Reserved	-	-	-	-	-
0xF0B3	Reserved	-		-	-	-
0xF0B4	SFR Guard Setting Register 0	SFRGD0L	SFRGD0	R/W	8/16	0x00
0xF0B5		SFRGD0H		R/W	8	0x00
0xF0B6	SFR Guard Setting Register 1	SFRGD1L	SFRGD1	R/W	8/16	0x00
0xF0B7		SFRGD1H		R/W	8	0x00
0xF0B8 to 0xFBB	Reserved	-	-	-	-	-
0xF0BC	RAM Parity Setting Register	RASFMOD	-	R/W	8	0x00
0xF0BD	Reserved	-		-	-	-
0xF0BE	Communication Test Setting Register 0	COMFT0L	COMFT0	R/W	8/16	0x00
0xF0BF		COMFT0H		R/W	8	0x00
0xF0C0 to 0xF1FF	Reserved	-	-	-	-	-
0xF200	Port 0 data register	P0DI	P0D	R	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	Port 0 mode register 01	P0MOD0	P0MOD01	R/W	8/16	0x05
0xF203		P0MOD1		R/W	8	0x00
0xF204	Port 0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x00
0xF205		P0MOD3		R/W	8	0x00
0xF206	Port 0 mode register 45	P0MOD4	P0MOD45	R/W	8/16	0x00
0xF207		P0MOD5		R/W	8	0x00
0xF208	Port 0 mode register 67	P0MOD6	P0MOD67	R/W	8/16	0x00
0xF209		P0MOD7		R/W	8	0x00
0xF20A	Reserved	-	-	-	-	-
0xF20B	Reserved	-		-	-	-
0xF20C	Reserved	-	-	-	-	-
0xF20D	Reserved	-		-	-	-
0xF20E	Reserved	-	-	-	-	-
0xF20F	Reserved	-		-	-	-
0xF210	Port 1 data register	P1DI	P1D	R	8/16	0xFF
0xF211		P1DO		R/W	8	0x00
0xF212	Port 1 mode register 01	P1MOD0	P1MOD01	R/W	8/16	0x00
0xF213		P1MOD1		R/W	8	0x00
0xF214	Port 1 mode register 23	P1MOD2	P1MOD23	R/W	8/16	0x00
0xF215		P1MOD3		R/W	8	0x00
0xF216	Port 1 mode register 45	P1MOD4	P1MOD45	R/W	8/16	0x00
0xF217		P1MOD5		R/W	8	0x00
0xF218	Port 1 mode register 67	P1MOD6	P1MOD67	R/W	8/16	0x00
0xF219		P1MOD7		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF21A	Port 1 pulse mode register	P1PMDL	P1PMD	R/W	8/16	0x00
0xF21B		P1PMDH		R/W	8	0x00
0xF21C	Port 1 pulse selection register	P1PSLL	P1PSL	R/W	8/16	0x00
0xF21D		P1PSLH		R/W	8	0x00
0xF21E	Reserved	-	-	-	-	-
0xF21F	Reserved	-		-	-	-
0xF220	Port 2 data register	P2DI	P2D	R	8/16	0xFF
0xF221		P2DO		R/W	8	0x00
0xF222	Port 2 mode register 01	P2MOD0	P2MOD01	R/W	8/16	0x00
0xF223		P2MOD1		R/W	8	0x00
0xF224	Port 2 mode register 23	P2MOD2	P2MOD23	R/W	8/16	0x00
0xF225		P2MOD3		R/W	8	0x00
0xF226 to 0xF22F	Reserved	-	-	-	-	-
0xF300	16-bit timer 0 data register	TMH0DL	TMH0D	R/W	8/16	0xFF
0xF301		TMH0DH		R/W	8	0xFF
0xF302	16-bit timer 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF303		TMH0CH		R/W	8	0x00
0xF304	16-bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF305		TMH0MODH		R/W	8	0x00
0xF306 to 0xF33F	Reserved	-	-	-	-	-
0xF340	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF341		TMHSTRH		W	8	0x00
0xF342	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00
0xF343		TMHSTPH		W	8	0x00
0xF344	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF345		TMHSTATH		R	8	0x00
0xF346 to 0xF37F	Reserved	-	-	-	-	-
0xF380	OTM common update register	OTCUD	-	W	8	0x00
0xF381	Reserved register	-		-	-	-
0xF382	OTM common control register	OTCCONL	OTCCON	R/W	8/16	0x00
0xF383		OTCCONH		R/W	8	0x00
0xF384	OTM common start register	OTCSTRL	OTCSTR	W	8/16	0x00
0xF385		OTCSTRH		W	8	0x00
0xF386	OTM common stop register	OTCSTPL	OTCSTP	W	8/16	0x00
0xF387		OTCSTPH		W	8	0x00
0xF388	OTM common status register	OTCSTATL	OTCSTAT	R	8/16	0x00
0xF389		OTCSTATH		R	8	0x00
0xF38A	OTM common break control register	OTCBRK	-	R/W	8	0x3F
0xF38B	Reserved register	-		-	-	-
0xF38C	OTM common OTMn0 output logical authorized register	OTC0OCONL	OTC0OCON	R/W	8/16	0x00
0xF38D		OTC0OCONH		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF38E	OTM common OTMn1 output logical authorized register	OTC1OCONL	OTC1OCON	R/W	8/16	0x00
0xF38F		OTC1OCONH		R/W	8	0x00
0xF390	OTM0 output logical authorized register	OT0OCONL	OT0OCON	R/W	8/16	0x00
0xF391		OT0OCONH		R/W	8	0x00
0xF392	OTM1 output logical authorized register	OT1OCONL	OT1OCON	R/W	8/16	0x00
0xF393		OT1OCONH		R/W	8	0x00
0xF394	OTM2 output logical authorized register	OT2OCONL	OT2OCON	R/W	8/16	0x00
0xF395		OT2OCONH		R/W	8	0x00
0xF396	OTM3 output logical authorized register	OT3OCONL	OT3OCON	R/W	8/16	0x00
0xF397		OT3OCONH		R/W	8	0x00
0xF398	OTM4 output logical authorized register	OT4OCONL	OT4OCON	R/W	8/16	0x00
0xF399		OT4OCONH		R/W	8	0x00
0xF39A	OTM5 output logical authorized register	OT5OCONL	OT5OCON	R/W	8/16	0x00
0xF39B		OT5OCONH		R/W	8	0x00
0xF39C	Reserved register	-	-	-	-	-
0xF39D		-		-	-	-
0xF39E	Reserved register	-	-	-	-	-
0xF39F		-		-	-	-
0xF3A0	OTM0 counter trigger 1 register	OT0CTRG1L	OT0CTRG1	R/W	8/16	0x00
0xF3A1		OT0CTRG1H		R/W	8	0x00
0xF3A2	OTM1 counter trigger 1 register	OT1CTRG1L	OT1CTRG1	R/W	8/16	0x00
0xF3A3		OT1CTRG1H		R/W	8	0x00
0xF3A4	OTM2 counter trigger 1 register	OT2CTRG1L	OT2CTRG1	R/W	8/16	0x00
0xF3A5		OT2CTRG1H		R/W	8	0x00
0xF3A6	OTM3 counter trigger 1 register	OT3CTRG1L	OT3CTRG1	R/W	8/16	0x00
0xF3A7		OT3CTRG1H		R/W	8	0x00
0xF3A8	OTM4 counter trigger 1 register	OT4CTRG1L	OT4CTRG1	R/W	8/16	0x00
0xF3A9		OT4CTRG1H		R/W	8	0x00
0xF3AA	OTM5 counter trigger 1 register	OT5CTRG1L	OT5CTRG1	R/W	8/16	0x00
0xF3AB		OT5CTRG1H		R/W	8	0x00
0xF3AC to 0xF3EF	Reserved register	-	-	-	-	-
0xF3F0	OTM Cycle setting method selection register	OTCPCL	OTCPC	R/W	8/16	0x00
0xF3F1		OTCPCH		R/W	8	0x00
0xF3F2 to 0xF3FF	Reserved register	-	-	-	-	-
0xF400	OTM0 cycle register	OT0PL	OT0P	R/W	8/16	0xFF
0xF401		OT0PH		R/W	8	0xFF
0xF402	OTM00 Rise Point register	OT00RL	OT00R	R/W	8/16	0x00
0xF403		OT00RH		R/W	8	0x00
0xF404	OTM00 Fall Point register	OT00FL	OT00F	R/W	8/16	0x00
0xF405		OT00FH		R/W	8	0x00
0xF406	OTM01 Rise Point register	OT01RL	OT01R	R/W	8/16	0x00
0xF407		OT01RH		R/W	8	0x00
0xF408	OTM01 Fall Point register	OT01FL	OT01F	R/W	8/16	0x00
0xF409		OT01FH		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF40A	OTM0 counter register	OT0CL	OT0C	R/W	8/16	0x00
0xF40B		OT0CH		R/W	8	0x00
0xF40C	OTM0 status register	OT0STATL	OT0STAT	R	8/16	0x06
0xF40D		OT0STATH		R	8	0x00
0xF40E	OTM0 mode register	OT0MODL	OT0MOD	R/W	8/16	0x00
0xF40F		OT0MODH		R/W	8	0x40
0xF410	OTM0 cycle stop timing register	OT0SCLRL	OT0SCLR	R/W	8/16	0x00
0xF411		OT0SCLRH		R/W	8	0x00
0xF412	OTM0 clock register	OT0CLKL	OT0CLK	R/W	8/16	0x00
0xF413		OT0CLKH		R/W	8	0x00
0xF414	OTM0 counter trigger 0 register	OT0CTRG0L	OT0CTRG0	R/W	8/16	0x00
0xF415		OT0CTRG0H		R/W	8	0x00
0xF416	OTM0 Forced stop trigger 0 register	OT0FTRG0L	OT0FTRG0	R/W	8/16	0x00
0xF417		OT0FTRG0H		R/W	8	0x00
0xF418	OTM0 Forced stop trigger 1 register	OT0FTRG1L	OT0FTRG1	R/W	8/16	0x00
0xF419		OT0FTRG1H		R/W	8	0x00
0xF41A	OTM0 Interrupt enable Register	OT0INTEL	OT0INTE	R/W	8/16	0x00
0xF41B		OT0INTEH		R/W	8	0x00
0xF41C	OTM0 interrupt status register	OT0INTSL	OT0INTS	R	8/16	0x00
0xF41D		OT0INTSH		R	8	0x00
0xF41E	OTM0 interrupt clear register	OT0INTCL	-	W	8	0x00
0xF41F		OT0INTCH		W	8	0x00
0xF420	OTM1 cycle register	OT1PL	OT1P	R/W	8/16	0xFF
0xF421		OT1PH		R/W	8	0xFF
0xF422	OTM10 Rise Point register	OT10RL	OT10R	R/W	8/16	0x00
0xF423		OT10RH		R/W	8	0x00
0xF424	OTM10 Fall Point register	OT10FL	OT10F	R/W	8/16	0x00
0xF425		OT10FH		R/W	8	0x00
0xF426	OTM11 Rise Point register	OT11RL	OT11R	R/W	8/16	0x00
0xF427		OT11RH		R/W	8	0x00
0xF428	OTM11 Fall Point register	OT11FL	OT11F	R/W	8/16	0x00
0xF429		OT11FH		R/W	8	0x00
0xF42A	OTM1 counter register	OT1CL	OT1C	R/W	8/16	0x00
0xF42B		OT1CH		R/W	8	0x00
0xF42C	OTM1 status register	OT1STATL	OT1STAT	R	8/16	0x06
0xF42D		OT1STATH		R	8	0x00
0xF42E	OTM1 mode register	OT1MODL	OT1MOD	R/W	8/16	0x00
0xF42F		OT1MODH		R/W	8	0x40
0xF430	OTM1 cycle stop timing register	OT1SCLRL	OT1SCLR	R/W	8/16	0x00
0xF431		OT1SCLRH		R/W	8	0x00
0xF432	OTM1 clock register	OT1CLKL	OT1CLK	R/W	8/16	0x00
0xF433		OT1CLKH		R/W	8	0x00
0xF434	OTM1 counter trigger 0 register	OT1CTRG0L	OT1CTRG0	R/W	8/16	0x00
0xF435		OT1CTRG0H		R/W	8	0x00
0xF436	OTM1 Forced stop trigger 0 register	OT1FTRG0L	OT1FTRG0	R/W	8/16	0x00
0xF437		OT1FTRG0H		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF438	OTM1 Forced stop trigger 1 register	OT1FTRG1L	OT1FTRG1	R/W	8/16	0x00
0xF439		OT1FTRG1H		R/W	8	0x00
0xF43A	OTM1 interrupt enable register	OT1INTEL	OT1INTE	R/W	8/16	0x00
0xF43B		OT1INTEH		R/W	8	0x00
0xF43C	OTM1 interrupt status register	OT1INTSL	OT1INTS	R	8/16	0x00
0xF43D		OT1INTSH		R	8	0x00
0xF43E	OTM1 interrupt clear register(L/H)	OT1INTCL	-	W	8	0x00
0xF43F		OT1INTCH		W	8	0x00
0xF440	OTM2 cycle register	OT2PL	OT2P	R/W	8/16	0xFF
0xF441		OT2PH		R/W	8	0xFF
0xF442	OTM20 Rise Point register	OT20RL	OT20R	R/W	8/16	0x00
0xF443		OT20RH		R/W	8	0x00
0xF444	OTM20 Fall Point register	OT20FL	OT20F	R/W	8/16	0x00
0xF445		OT20FH		R/W	8	0x00
0xF446	Reserved register	-	-	-	-	-
0xF447		-		-	-	-
0xF448	Reserved register	-	-	-	-	-
0xF449		-		-	-	-
0xF44A	OTM2 counter register	OT2CL	OT2C	R/W	8/16	0x00
0xF44B		OT2CH		R/W	8	0x00
0xF44C	OTM2 status register	OT2STATL	OT2STAT	R	8/16	0x02
0xF44D		OT2STATH		R	8	0x00
0xF44E	OTM2 mode register	OT2MODL	OT2MOD	R/W	8/16	0x00
0xF44F		OT2MODH		R/W	8	0x00
0xF450	OTM2 cycle stop timing register	OT2SCLRL	OT2SCLR	R/W	8/16	0x00
0xF451		OT2SCLRH		R/W	8	0x00
0xF452	OTM2 clock register	OT2CLKL	OT2CLK	R/W	8/16	0x00
0xF453		OT2CLKH		R/W	8	0x00
0xF454	OTM2 counter trigger 0 register	OT2CTRG0L	OT2CTRG0	R/W	8/16	0x00
0xF455		OT2CTRG0H		R/W	8	0x00
0xF456	OTM2 Forced stop trigger 0 register	OT2FTRG0L	OT2FTRG0	R/W	8/16	0x00
0xF457		OT2FTRG0H		R/W	8	0x00
0xF458	OTM2 Forced stop trigger 1 register	OT2FTRG1L	OT2FTRG1	R/W	8/16	0x00
0xF459		OT2FTRG1H		R/W	8	0x00
0xF45A	OTM2 interrupt enable register	OT2INTEL	OT2INTE	R/W	8/16	0x00
0xF45B		OT2INTEH		R/W	8	0x00
0xF45C	OTM2 interrupt status register	OT2INTSL	OT2INTS	R	8/16	0x00
0xF45D		OT2INTSH		R	8	0x00
0xF45E	OTM2 interrupt clear register(L/H)	OT2INTCL	-	W	8	0x00
0xF45F		OT2INTCH		W	8	0x00
0xF460	OTM3 cycle register	OT3PL	OT3P	R/W	8/16	0xFF
0xF461		OT3PH		R/W	8	0xFF
0xF462	OTM30 Rise Point register	OT30RL	OT30R	R/W	8/16	0x00
0xF463		OT30RH		R/W	8	0x00
0xF464	OTM30 Fall Point register	OT30FL	OT30F	R/W	8/16	0x00
0xF465		OT30FH		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF466	Reserved register	-	-	-	-	-
0xF467		-		-	-	-
0xF468	Reserved register	-	-	-	-	-
0xF469		-		-	-	-
0xF46A	OTM3 counter register	OT3CL	OT3C	R/W	8/16	0x00
0xF46B		OT3CH		R/W	8	0x00
0xF46C	OTM3 status register	OT3STATL	OT3STAT	R	8/16	0x02
0xF46D		OT3STATH		R	8	0x00
0xF46E	OTM3 mode register	OT3MODL	OT3MOD	R/W	8/16	0x00
0xF46F		OT3MODH		R/W	8	0x00
0xF470	OTM3 cycle stop timing register	OT3SCLRL	OT3SCLR	R/W	8/16	0x00
0xF471		OT3SCLRH		R/W	8	0x00
0xF472	OTM3 clock register	OT3CLKL	OT3CLK	R/W	8/16	0x00
0xF473		OT3CLKH		R/W	8	0x00
0xF474	OTM3 counter trigger 0 register	OT3CTRG0L	OT3CTRG0	R/W	8/16	0x00
0xF475		OT3CTRG0H		R/W	8	0x00
0xF476	OTM3 Forced stop trigger 0 register	OT3FTRG0L	OT3FTRG0	R/W	8/16	0x00
0xF477		OT3FTRG0H		R/W	8	0x00
0xF478	OTM3 Forced stop trigger 1 register	OT3FTRG1L	OT3FTRG1	R/W	8/16	0x00
0xF479		OT3FTRG1H		R/W	8	0x00
0xF47A	OTM3 interrupt enable register	OT3INTEL	OT3INTE	R/W	8/16	0x00
0xF47B		OT3INTEH		R/W	8	0x00
0xF47C	OTM3 interrupt status register	OT3INTSL	OT3INTS	R	8/16	0x00
0xF47D		OT3INTSH		R	8	0x00
0xF47E	OTM3 interrupt clear register(L/H)	OT3INTCL	-	W	8	0x00
0xF47F		OT3INTCH		W	8	0x00
0xF480	OTM4 cycle register	OT4PL	OT4P	R/W	8/16	0xFF
0xF481		OT4PH		R/W	8	0xFF
0xF482	OTM40 Rise Point register	OT40RL	OT40R	R/W	8/16	0x00
0xF483		OT40RH		R/W	8	0x00
0xF484	OTM40 Fall Point register	OT40FL	OT40F	R/W	8/16	0x00
0xF485		OT40FH		R/W	8	0x00
0xF486	OTM41 Rise Point register	OT41RL	OT41R	R/W	8/16	0x00
0xF487		OT41RH		R/W	8	0x00
0xF488	OTM41 Fall Point register	OT41FL	OT41F	R/W	8/16	0x00
0xF489		OT41FH		R/W	8	0x00
0xF48A	OTM4 counter register	OT4CL	OT4C	R/W	8/16	0x00
0xF48B		OT4CH		R/W	8	0x00
0xF48C	OTM4 status register	OT4STATL	OT4STAT	R	8/16	0x06
0xF48D		OT4STATH		R	8	0x00
0xF48E	OTM4 mode register	OT4MODL	OT4MOD	R/W	8/16	0x00
0xF48F		OT4MODH		R/W	8	0x40
0xF490	OTM4 cycle stop timing register	OT4SCLRL	OT4SCLR	R/W	8/16	0x00
0xF491		OT4SCLRH		R/W	8	0x00
0xF492	OTM4 clock register	OT4CLKL	OT4CLK	R/W	8/16	0x00
0xF493		OT4CLKH		R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF494	OTM4 counter trigger 0 register	OT4CTRG0L	OT4CTRG0	R/W	8/16	0x00
0xF495		OT4CTRG0H		R/W	8	0x00
0xF496	OTM4 Forced stop trigger 0 register	OT4FTRG0L	OT4FTRG0	R/W	8/16	0x00
0xF497		OT4FTRG0H		R/W	8	0x00
0xF498	OTM4 Forced stop trigger 1 register	OT4FTRG1L	OT4FTRG1	R/W	8/16	0x00
0xF499		OT4FTRG1H		R/W	8	0x00
0xF49A	OTM4 interrupt enable register	OT4INTEL	OT4INTE	R/W	8/16	0x00
0xF49B		OT4INTEH		R/W	8	0x00
0xF49C	OTM4 interrupt status register	OT4INTSL	OT4INTS	R	8/16	0x00
0xF49D		OT4INTSH		R	8	0x00
0xF49E	OTM4 interrupt clear register(L/H)	OT4INTCL	-	W	8	0x00
0xF49F		OT4INTCH		W	8	0x00
0xF4A0	OTM5 cycle register	OT5PL	OT5P	R/W	8/16	0xFF
0xF4A1		OT5PH		R/W	8	0xFF
0xF4A2	OTM50 Rise Point register	OT50RL	OT50R	R/W	8/16	0x00
0xF4A3		OT50RH		R/W	8	0x00
0xF4A4	OTM50 Fall Point register	OT50FL	OT50F	R/W	8/16	0x00
0xF4A5		OT50FH		R/W	8	0x00
0xF4A6	OTM51 Rise Point register	OT51RL	OT51R	R/W	8/16	0x00
0xF4A7		OT51RH		R/W	8	0x00
0xF4A8	OTM51 Fall Point register	OT51FL	OT51F	R/W	8/16	0x00
0xF4A9		OT51FH		R/W	8	0x00
0xF4AA	OTM5 counter register	OT5CL	OT5C	R/W	8/16	0x00
0xF4AB		OT5CH		R/W	8	0x00
0xF4AC	OTM5 status register	OT5STATL	OT5STAT	R	8/16	0x06
0xF4AD		OT5STATH		R	8	0x00
0xF4AE	OTM5 mode register	OT5MODL	OT5MOD	R/W	8/16	0x00
0xF4AF		OT5MODH		R/W	8	0x40
0xF4B0	OTM5 cycle stop timing register	OT5SCLRL	OT5SCLR	R/W	8/16	0x00
0xF4B1		OT5SCLRH		R/W	8	0x00
0xF4B2	OTM5 clock register	OT5CLKL	OT5CLK	R/W	8/16	0x00
0xF4B3		OT5CLKH		R/W	8	0x00
0xF4B4	OTM5 counter trigger 0 register	OT5CTRG0L	OT5CTRG0	R/W	8/16	0x00
0xF4B5		OT5CTRG0H		R/W	8	0x00
0xF4B6	OTM5 Forced stop trigger 0 register	OT5FTRG0L	OT5FTRG0	R/W	8/16	0x00
0xF4B7		OT5FTRG0H		R/W	8	0x00
0xF4B8	OTM5 Forced stop trigger 1 register	OT5FTRG1L	OT5FTRG1	R/W	8/16	0x00
0xF4B9		OT5FTRG1H		R/W	8	0x00
0xF4BA	OTM5 interrupt enable register	OT5INTEL	OT5INTE	R/W	8/16	0x00
0xF4BB		OT5INTEH		R/W	8	0x00
0xF4BC	OTM5 interrupt status register	OT5INTSL	OT5INTS	R	8/16	0x00
0xF4BD		OT5INTSH		R	8	0x00
0xF4BE	OTM5 interrupt clear register(L/H)	OT5INTCL	-	W	8	0x00
0xF4BF		OT5INTCH		W	8	0x00
0xF4C0 to 0xF5FF	Reserved	-	-	-	-	-

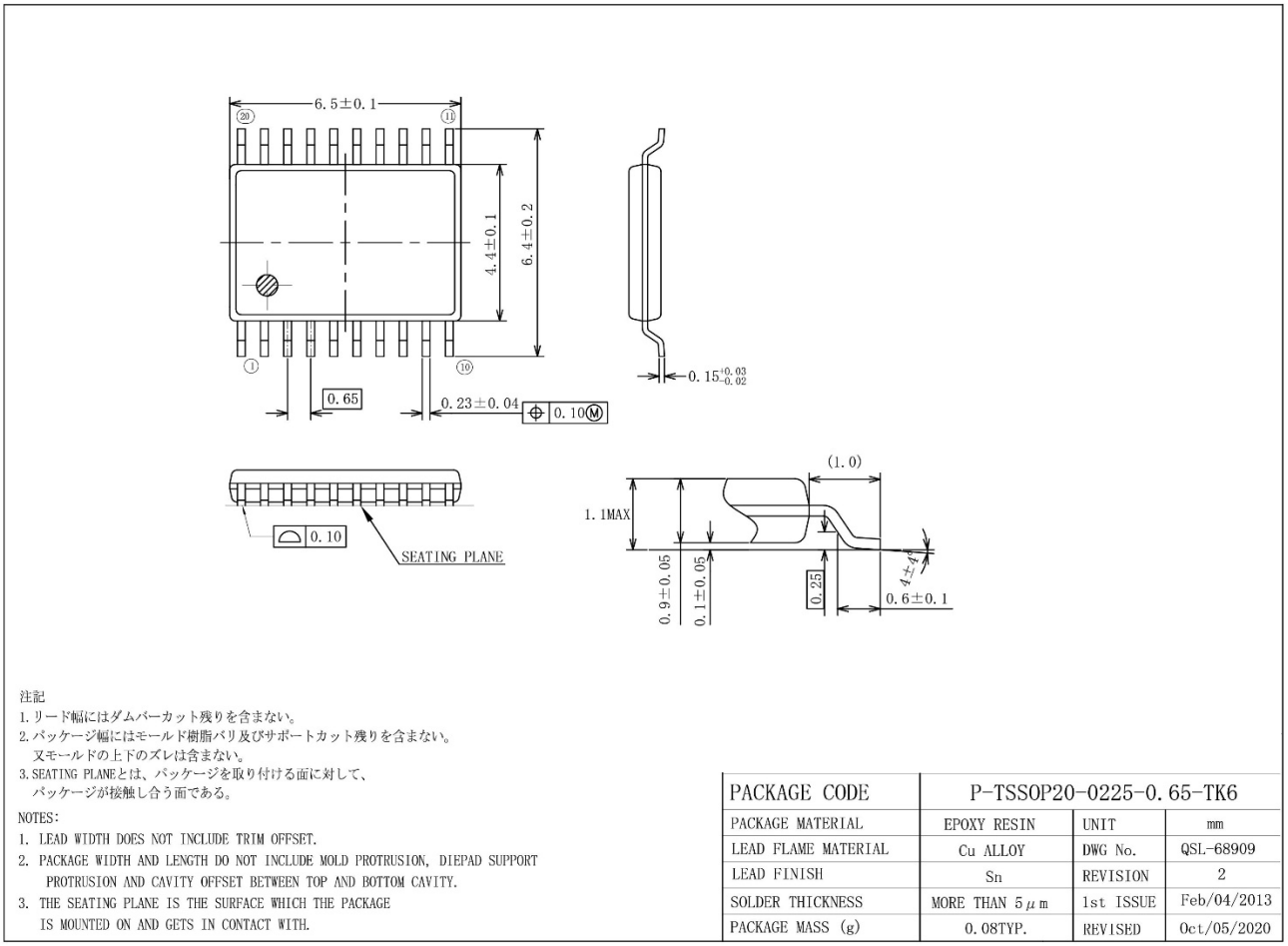
Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF600	UART0 reception buffer	UA0BUF0	-	R	8	0x00
0xF601	UART0 transmission buffer	UA0BUF1		R/W	8	0x00
0xF602	UART0 status register	UA0STAT	-	R	8	0x00
0xF603	UART0 status clear register	UA0STAC		W	8	0x00
0xF604	UART0 control register	UA0CON	-	R/W	8	0x00
0xF605	Reserved	-		-	-	-
0xF606	UART0 mode register	UA0MODL	UA0MOD	R/W	8/16	0x01
0xF607		UA0MODH		R/W	8	0x00
0xF608	UART0 interrupt enable register	UA0INTE	-	R/W	8	0x00
0xF609	Reserved	-		-	-	-
0xF60A	UART0 baud rate register	UA0BRTL	UA0BRT	R/W	8/16	0xFF
0xF60B		UA0BRTH		R/W	8	0x0F
0xF60C	UART0 baud rate adjustment register	UA0BRC	-	R/W	8	0x00
0xF60D	Reserved	-		-	-	-
0xF60E	Reserved	-	-	-	-	-
0xF60F	Reserved	-		-	-	-
0xF610	UART1 reception buffer	UA1BUF0	-	R	8	0x00
0xF611	UART1 transmission buffer	UA1BUF1		R/W	8	0x00
0xF612	UART1 status register	UA1STAT	-	R	8	0x00
0xF613	UART1 status clear register	UA1STAC		W	8	0x00
0xF614	UART1 control register	UA1CON	-	R/W	8	0x00
0xF615	Reserved	-		-	-	-
0xF616	UART1 mode register	UA1MODL	UA1MOD	R/W	8/16	0x01
0xF617		UA1MODH		R/W	8	0x00
0xF618	UART1 interrupt enable register	UA1INTE	-	R/W	8	0x00
0xF619	Reserved	-		-	-	-
0xF61A	UART1 baud rate register	UA1BRTL	UA1BRT	R/W	8/16	0xFF
0xF61B		UA1BRTH		R/W	8	0x0F
0xF61C	UART1 baud rate adjustment register	UA1BRC	-	R/W	8	0x00
0xF61D	Reserved	-		-	-	-
0xF61E to 0xF77F	Reserved	-	-	-	-	-
0xF780	I^2C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF781	Reserved	-		-	-	-
0xF782	I^2C bus unit 0 receive register	I2U0RD	-	R	8	0x00
0xF783	Reserved	-		-	-	-
0xF784	I^2C bus unit 0 slave address register	I2U0SA	-	R/W	8	0x00
0xF785	Reserved	-		-	-	-
0xF786	I^2C bus unit 0 transmit data register	I2U0TD	-	R/W	8	0x00
0xF787	Reserved	-		-	-	-
0xF788	I^2C bus unit 0 control register	I2U0CON	-	R/W	8	0x00
0xF789	Reserved	-		-	-	-
0xF78A	I^2C bus unit 0 mode register	I2U0MODL	I2U0MOD	R/W	8/16	0x00
0xF78B		I2U0MODH		R/W	8	0x22
0xF78C	I^2C bus unit 0 status register	I2U0STAT	I2U0STR	R	8/16	0x00
0xF78D		I2U0ISR		R	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF78E	I^2C bus unit 0 status clear register	I2U0SCLRL	I2U0SCLR	W	8/16	0x00
0xF78F		I2U0SCLRH		W	8	0x00
0xF790 to 0xF7FF	Reserved	-	-	-	-	-
0xF800	SA-ADC mode register	SADMODL	SADMOD	R/W	8/16	0x30
0xF801		SADMODH		R/W	8	0x00
0xF802	SA-ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF803		SADCONH		R/W	8	0x00
0xF804	SA-ADC conversion interval register	SADSTML	SADSTM	R/W	8/16	0x00
0xF805		SADSTMH		R/W	8	0x00
0xF806	Reference voltage control register	VREFCON	-	R/W	8	0x00
0xF807	Reserved	-		-	-	-
0xF808	SA-ADC interrupt mode register	SADIMOD	-	R/W	8	0x00
0xF809	Reserved	-		-	-	-
0xF80A	SA-ADC trigger register	SADTRG	-	R/W	8	0x00
0xF80B	Reserved	-		-	-	-
0xF80C	SA-ADC enable register 0	SADEN0L	SADEN0	R/W	8/16	0x00
0xF80D		SADEN0H		R/W	8	0x00
0xF80E	SA-ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF80F		SADEN1H		R/W	8	0x00
0xF810 to 0xF82F	Reserved	-	-	-	-	-
0xF830	SA-ADC test mode register	SADTMOD	-	R/W	8	0x00
0xF831	Reserved	-		-	-	-
0xF832 to 0xF87F	Reserved	-	-	-	-	-
0xF880	Comparator control register	CMPCONL	CMPCON	R/W	8/16	0x00
0xF881		CMPCONH		R/W	8	0x00
0xF882	Comparator 0 mode register	CMP0MODL	CMP0MOD	R/W	8/16	0x00
0xF883		CMP0MODH		R/W	8	0x00
0xF884	Comparator 1 mode register	CMP1MODL	CMP1MOD	R/W	8/16	0x00
0xF885		CMP1MODH		R/W	8	0x00
0xF886	Comparator 2 mode register	CMP2MODL	CMP2MOD	R/W	8/16	0x00
0xF887		CMP2MODH		R/W	8	0x00
0xF888	Comparator status register	CMPSTA	-	R	8	0x00
0xF889	Reserved register	-		R	8	0x00
0xF88A to 0xF88F	Reserved register	-	-	R	8	0x00
0xF890	Low Level Detector control register	LLD0CON	-	R/W	8	0x00
0xF891	Reserved	-		-	-	-
0xF892	Low Level Detector mode register	LLD0MOD	-	R/W	8	0x00
0xF893	Reserved	-		-	-	-
0xF894	Reserved	-	-	-	-	-
0xF895	Reserved	-		-	-	-
0xF896	Low Level Detector sampling register	LLD0SMP	-	R/W	8	0x00

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF897	Reserved	-		-	-	-
0xF898 to 0xF89F	Reserved	-	-	-	-	-
0xF8A0	D/A converter control register	DACCON	-	R/W	8	0x00
0xF8A1	Reserved	-		-	-	-
0xF8A2	D/A converter mode register	DACMODE	-	R/W	8	0x00
0xF8A3	Reserved	-		-	-	-
0xF8A4	D/A converter 0 code register	DAC0CODE	-	R/W	8	0x00
0xF8A5	Reserved	-		-	-	-
0xF8A6	D/A converter 1 code register	DAC1CODE	-	R/W	8	0x00
0xF8A7	Reserved	-		-	-	-
0xF8A8 to 0xF8AF	Reserved	-	-	-	-	-
0xF8B0	PGA control register	PGACON	-	R/W	8	0x00
0xF8B1	Reserved register	-		-	-	-
0xF8B2	PGA mode register	PGAMOD	-	R/W	8	0x00
0xF8B3	Reserved register	-		-	-	-
0xF8B4 to 0xF91F	Reserved register	-	-	-	-	-
0xF920	Code Option 0	CODEOP0L	CODEOP0	R	8/16	*1
0xF921		CODEOP0H		R	8	*1
0xF922	Code Option 1	CODEOP1L	CODEOP1	R	8/16	*1
0xF923		CODEOP1H		R	8	*1
0xF924	Code Option 2	CODEOP2L	CODEOP2	R	8/16	*1
0xF925		CODEOP2H		R	8	*1
0xF926 to 0xF92F	Reserved register	-	-	-	-	-
0xF930	Product ID register 0	PID0L	PID0	R	8/16	*2
0xF931		PID0H		R	8	*2
0xF932	Product ID register 1	PID1L	PID1	R	8/16	0x22
0xF933		PID1H		R	8	0x06
0xF926 to 0xFFFF	Reserved register	-	-	-	-	-

Appendix B Package Dimensions

●20pin TSSOP



注記
1. リード幅にはダムバーカット残りを含まない。
2. パッケージ幅にはモールド樹脂バリ及びサポートカット残りを含まない。
又モールドの上下のズレは含まない。
3. SEATING PLANEとは、パッケージを取り付ける面に対して、
パッケージが接触し合う面である。

NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

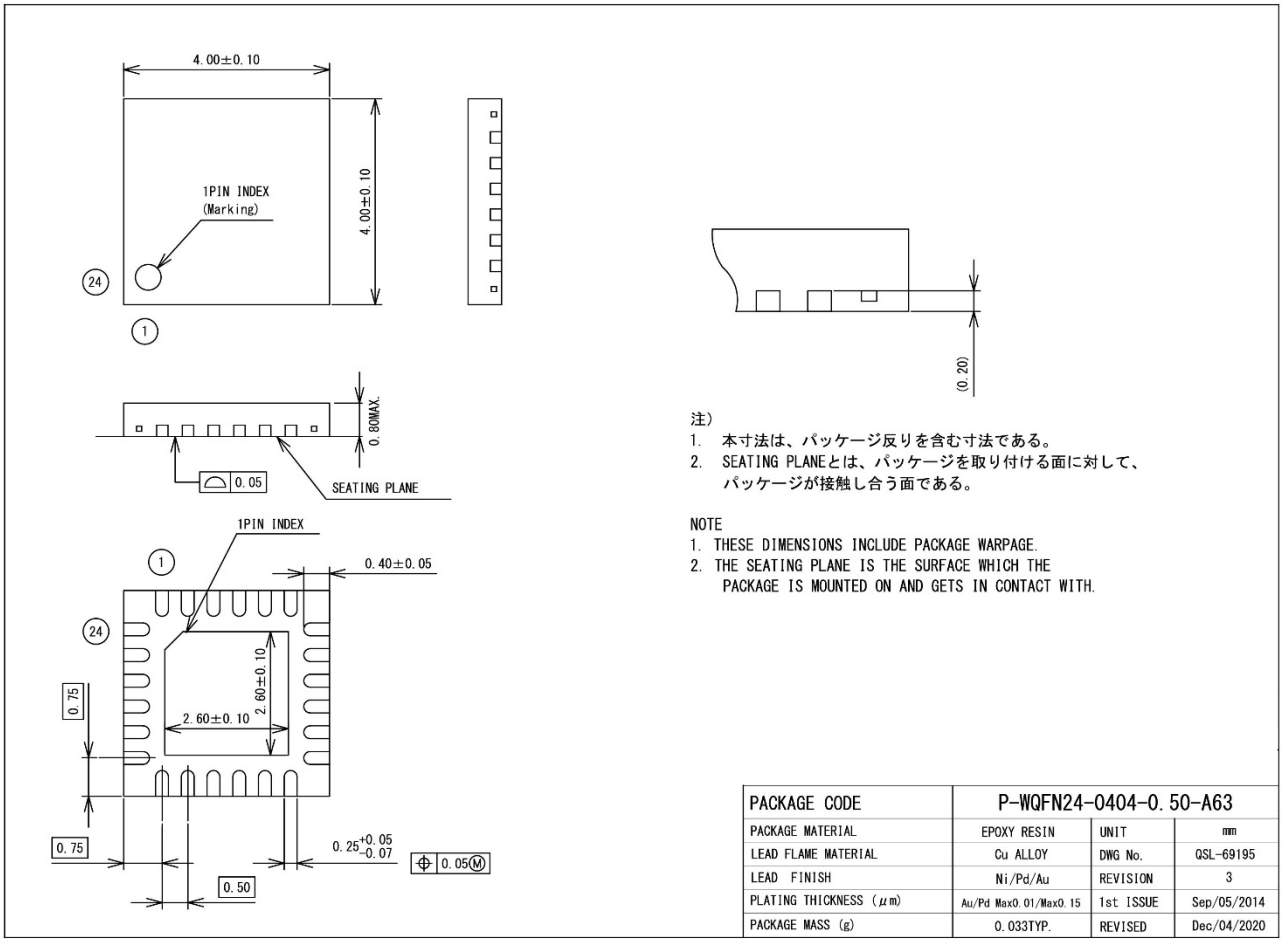
PACKAGE CODE	P-TSSOP20-0225-0.65-TK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-68909
LEAD FINISH	Sn	REVISION	2
SOLDER THICKNESS	MORE THAN 5μm	1st ISSUE	Feb/04/2013
PACKAGE MASS (g)	0.08TYP.	REVISED	Oct/05/2020

（Unit: mm）

Figure B-1 TSSOP20 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package
The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

●24pin WQFN



PACKAGE CODE	P-WQFN24-0404-0.50-A63		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69195
LEAD FINISH	Ni/Pd/Au	REVISION	3
PLATING THICKNESS (μm)	Au/Pd Max0.01/Max0.15	1st ISSUE	Sep/05/2014
PACKAGE MASS (g)	0.033TYP.	REVISED	Dec/04/2020

（Unit: mm）

Figure B-2 WQFN24 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package
The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

[Note] Notes for the package with exposed die pad
The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

Appendix C Instruction Execution Cycle

Tables on following pages show the all instructions of nX-U16/100 core and the execution cycles. "-" indicates that there is no memory access during the instruction execution. See "Example of Instruction execution cycle" for details on how to read the table.

Example of Instruction execution cycle

(1)			(2)	(3)	(4)	(5)
Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
ADD	ERn	ERm	1	-	-	-
B	Cadr		2	-	-	1
	ERn		2	-	-	1
L	ERn	[EA]	1	1	1	-
		[EA+]	1	1	1	-

[How to read the table]

1) These are the instructions of nX-U16/100 (A35 core)
2) The execution cycle of each instruction.
3) Additional execution cycle when the instruction refers to ROM.
4) Additional execution cycle when the instruction reads the address allocated in segment 1 or larger.
   One cycle is added in spite of the CPU operating mode.
   For more details, see the section 1.3.4 "DSR Prefix Instructions" in the nX-U16/100 core instruction manual.
5) Additional execution effected by the instruction with the [EA+] addressing.
   For more details, see the section 3.3 "Instruction Execution Times" in the nX-U16/100 core instruction manual.

Arithmetic Instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
ADD	ER*n*	ER*m*	1	-	-	-
		#imm7	1	-	-	-
ADD	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
ADDC	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
AND	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
CMP	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
CMPC	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
MOV	ER*n*	ER*m*	1	-	-	-
		#imm7	1	-	-	-
MOV	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
OR	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
XOR	R*n*	R*m*	1	-	-	-
		#imm8	1	-	-	-
CMP	ER*n*	ER*m*	1	-	-	-
SUB	R*n*	R*m*	1	-	-	-
SUBC	R*n*	R*m*	1	-	-	-

Shift instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
SLL	R*n*	R*m*	1	-	-	1
		#width	1	-	-	1
SLLC	R*n*	R*m*	1	-	-	1
		#width	1	-	-	1
SRA	R*n*	R*m*	1	-	-	1
		#width	1	-	-	1
SRL	R*n*	R*m*	1	-	-	1
		#width	1	-	-	1
SRLC	R*n*	R*m*	1	-	-	1
		#width	1	-	-	1

Load/Store instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
L	ER*n*	[EA]	1	1	1	-
		[EA+]	1	1	1	-
		[ERm]	1	1	1	1
		Disp16[ER*m*]	2	1	1	1
		Disp6[BP]	2	1	1	1
		Disp6[FP]	2	1	1	1
		Dadr	2	1	1	1
	R*n*	[EA]	1	1	1	-
		[EA+]	1	1	1	-
		[ER*m*]	1	1	1	1
		Disp16[ER*m*]	2	1	1	1
		Disp6[BP]	2	1	1	1
		Disp6[FP]	2	1	1	1
		Dadr	2	1	1	1
	XR*n*	[EA]	2	2	1	-
		[EA+]	2	2	1	-
	QR*n*	[EA]	4	4	1	-
		[EA+]	4	4	1	-
ST	ER*n*	[EA]	1	-	-	-
		[EA+]	1	-	-	-
		[ER*m*]	1	-	-	1
		Disp16[ER*m*]	2	-	-	1
		Disp6[BP]	2	-	-	1
		Disp6[FP]	2	-	-	1
		Dadr	2	-	-	1
	R*n*	[EA]	1	-	-	-
		[EA+]	1	-	-	-
		[ER*m*]	1	-	-	1
		Disp16[ER*m*]	2	-	-	1
		Disp6[BP]	2	-	-	1
		Disp6[FP]	2	-	-	1
		Dadr	2	-	-	1
	XR*n*	[EA]	2	-	-	-
		[EA+]	2	-	-	-
	QR*n*	[EA]	4	-	-	-
		[EA+]	4	-	-	-

(*1) When the immediately preceding instruction is for reading the data memory or not (not the instruction for reading the data memory / the instruction for reading the data memory)

Control Register Access Instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
ADD	SP	*#signed8*	1	-	-	-
MOV	ECSR	R*m*	1	-	-	-
	ELR	ER*m*	1	-	-	-
	EPSW	R*m*	1	-	-	-
	ER*n*	ELR	1	-	-	-
		SP	1	-	-	-
	PSW	R*m*	1	-	-	-
		#unsigned8	1	-	-	-
	R*n*	CR*m*	1	-	-	-
		ECSR	1	-	-	-
		EPSW	1	-	-	-
		PSW	1	-	-	-
	SP	ER*m*	1	-	-	-

PUSH/POP Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
PUSH	EA	1	-	-	1
	ELR	1 / 2 (*1)	-	-	1
	EA,ELR	2 / 3 (*1)	-	-	1
	EPSW	1	-	-	1
	EPSW,EA	2	-	-	1
	EPSW,ELR	2 / 3 (*1)	-	-	1
	EPSW,ELR, EA	3 / 4 (*1)	-	-	1
	LR	1 / 2 (*1)	-	-	1
	LR,EA	2 / 3 (*1)	-	-	1
	LR,ELR	2 / 4 (*1)	-	-	1
	LR,EA,ELR	3 / 5 (*1)	-	-	1
	LR,EPSW	2 / 3 (*1)	-	-	1
	LR,EPSW,EA	3 / 4 (*1)	-	-	1
	LR,EPSW,ELR	3 / 5 (*1)	-	-	1
	LR,ELR,EPSW,EA	4 / 6 (*1)	-	-	1
	ER*n*	1	-	-	1
	QR*n*	4	-	-	1
	R*n*	1	-	-	1
	XR*n*	2	-	-	1
POP	EA	2	-	-	1
	EA,LR	3 / 4 (*1)	-	-	1
	EA,PC	5 / 6 (*1)	-	-	1
	EA,PC,LR	6 / 8 (*1)	-	-	1
	EA,PC,PSW	6 / 7 (*1)	-	-	1
	EA,PC,PSW,LR	7 / 9 (*1)	-	-	1
	EA,PSW	3	-	-	1
	EA,PSW,LR	4 / 5 (*1)	-	-	1
	LR	1 / 2 (*1)	-	-	1
	LR,PSW	2 / 3 (*1)	-	-	1
	PC	3 / 4 (*1)	-	-	1
	PC,LR	4 / 6 (*1)	-	-	1
	PC,PSW	4 / 5 (*1)	-	-	1
	PC,PSW,LR	5 / 7 (*1)	-	-	1
	PSW	1	-	-	1
	ER*n*	1	-	-	1
	QR*n*	4	-	-	1
	R*n*	1	-	-	1
	XR*n*	2	-	-	1

(*1) When the memory mode is SMALL or LARGE (SMALL model/LARGE model)

Coprocessor Data Transfer Instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
MOV	CR*n*	R*m*	1	-	-	-
	CER*n*	[EA]	1	1	1	1
		[EA+]	1	1	1	1
	CQR*n*	[EA]	4	4	1	1
		[EA+]	4	4	1	1
	CR*n*	[EA]	1	1	1	1
		[EA+]	1	1	1	1
	CXR*n*	[EA]	2	2	1	1
		[EA+]	2	2	1	1
MOV	R*n*	CR*m*	1	-	-	-
	[EA]	CER*m*	1	1	1	1
	[EA+]	CER*m*	1	1	1	1
	[EA]	CQR*m*	4	4	1	1
	[EA+]	CQR*m*	4	4	1	1
	[EA]	CR*m*	1	1	1	1
	[EA+]	CR*m*	1	1	1	1
	[EA]	CXR*m*	2	2	1	1
	[EA+]	CXR*m*	2	2	1	1

EA Register Data Transfer Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
LEA	[ER*m*]	1	-	-	-
	Disp16[ER*m*]	2	-	-	-
	Dadr	2	-	-	-

ALU Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
DAA	R*n*	1	-	-	-
DAS	R*n*	1	-	-	-
NEG	R*n*	1	-	-	-

Bit Access Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
SB	D*bitadr*	2	-	1	-
	R*n.bit_offset*	1	-	-	-
RB	D*bitadr*	2	-	1	-
	R*n.bit_offset*	1	-	-	-
TB	D*bitadr*	2	1	1	-
	R*n.bit_offset*	1	-	-	-

PSW Access Instructions

Instruction (No wait mode)	Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
EI	1	-	-	-
DI	3	-	-	-
SC	1	-	-	-
RC	1	-	-	-
CPLC	1	-	-	-

Sign Extension Instruction

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
EXTBW	ER*n*	1	-	-	-

Branch Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
B	*Cadr*	2	-	-	1
	ER*n*	2	-	-	1
BL	*Cadr*	2	-	-	1
	ER*n*	2	-	-	1

(*1) When the immediately preceding instruction is for reading the data memory or not (not the instruction for reading the data memory / the instruction for reading the data memory)

Conditional Relative Branch Instructions

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
BGE	*Radr*	1 / 2(*1)	-	-	1
BLT	*Radr*	1 / 2(*1)	-	-	1
BGT	*Radr*	1 / 2(*1)	-	-	1
BLE	*Radr*	1 / 2(*1)	-	-	1
BGES	*Radr*	1 / 2(*1)	-	-	1
BLTS	*Radr*	1 / 2(*1)	-	-	1
BGTS	*Radr*	1 / 2(*1)	-	-	1
BLES	*Radr*	1 / 2(*1)	-	-	1
BNE	*Radr*	1 / 2(*1)	-	-	1
BEQ	*Radr*	1 / 2(*1)	-	-	1
BNV	*Radr*	1 / 2(*1)	-	-	1
BOV	*Radr*	1 / 2(*1)	-	-	1
BPS	*Radr*	1 / 2(*1)	-	-	1
BNS	*Radr*	1 / 2(*1)	-	-	1
BAL	*Radr*	2	-	-	1

(*1) When the branch condition is matched or not (Not matched / Matched)

Multiplication and Division Instructions

Instruction (No wait mode)			Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
MUL	ER*n*	R*m*	9	-	-	-
DIV	ER*n*	R*m*	17	-	-	-

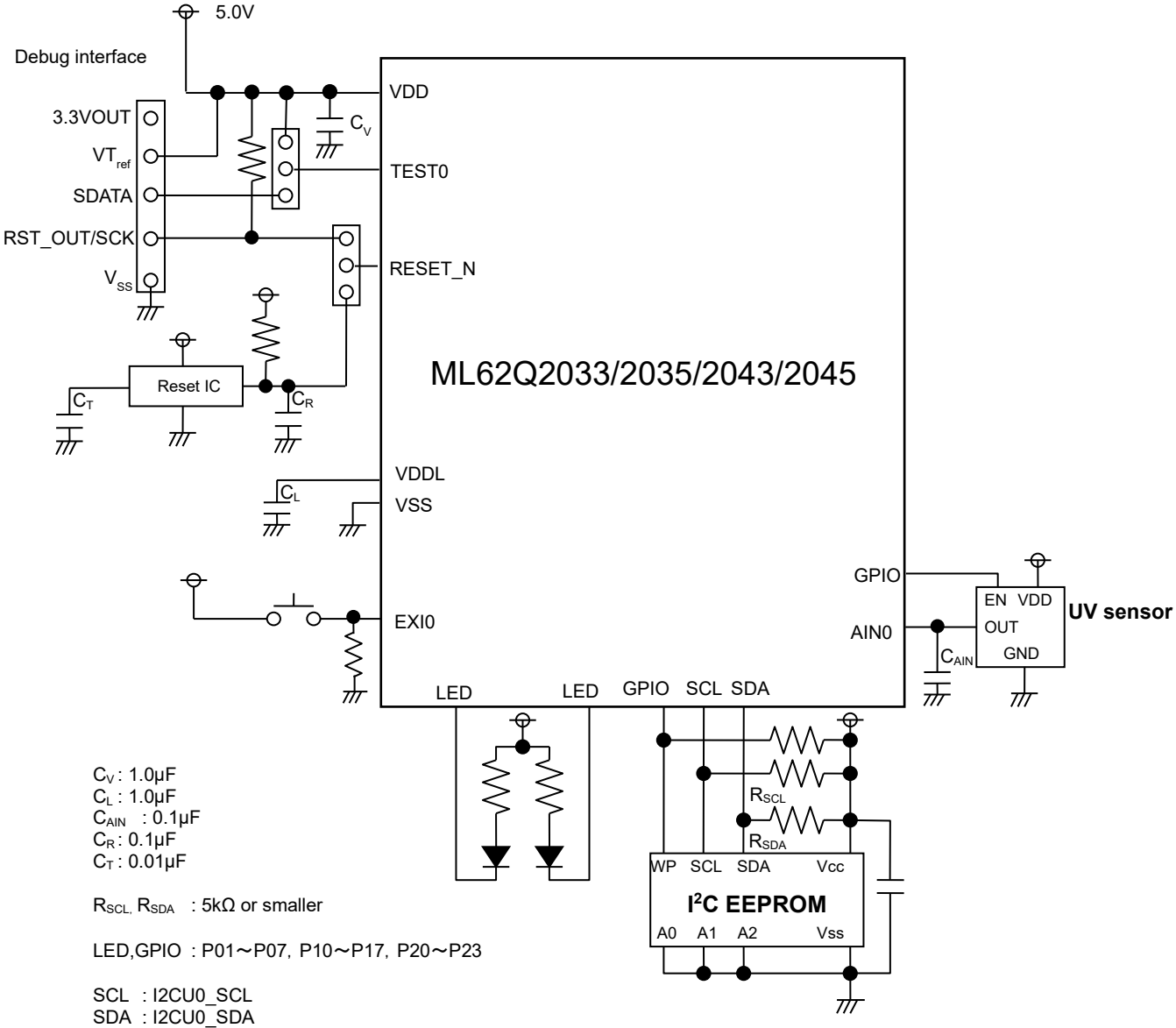
Interrupts

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
SWI	#*snum*	3	-	-	1
BRK		7	-	-	1
Interrupt transfer cycle		3	-	-	1

Miscellaneous

Instruction (No wait mode)		Min. execution cycle	ROM reference cycle	Effect of DSR access	Effect of [EA+] addressing
NOP		1	-	-	-
DEC	[EA]	2	-	1	1
INC	[EA]	2	-	1	1
RT		2	-	1	1
RTI		2	-	1	1

Appendix D Application Circuit Example



C_V : 1.0µF
C_L : 1.0µF
C_{AIN} : 0.1µF
C_R : 0.1µF
C_T : 0.01µF

R_{SCL}, R_{SDA} : 5kΩ or smaller

LED,GPIO : P01～P07, P10～P17, P20～P23

SCL : I2CU0_SCL
SDA : I2CU0_SDA

Reset IC : BU4217 (ROHM, Nch open drain output)

[Note]

- **Place the capacitor for VDDL pin as close to the LSI power pins as possible.**

Appendix E List of Notes

This Check List has important notes to prevent commonly-made programming mistakes and frequently overlooked or misunderstood hardware specifications of the LSI. Check each note listed in chapter by chapter when coding or evaluating the program.
Number in a mark [ ] shows section number to refer.

Common to all Chapters

- □ Please see the "Notes" and the "Notes for product usage" in this document front pages.
- □ Word access is available for registers with the word symbol. Specify an even address for the word access. See "List of Registers" in each chapter.
- □ Registers for unequipped channels are not available to use. They return 0x0000 for reading. See "List of Registers" in each chapter.

1. Overview

- □ [1.3.4] Terminate unused input pins according to the Table 1-5 in order to avoid unexpected through-current in the pins.

2. CPU and Memory Space

- □ [2.3.3] □ Changing to STOP/HALT mode during calculation is prohibited. When changing to STOP/HALT mode after using the multiplier and divider, wait until the calculation time in Table 2-7 has elapsed.
- □ [2.5] The CSR of this LSI is fixed at "0".
- □ [2.5] The Code Option area (48 bytes) is not available for the program code area. For details of Code Option settings, see Chapter 30 "Code Option" and make sure the setting data is correct.
- □ [2.5] It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using the generation tool of the ROM code data. See its manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.
- □ [2.5, 2.6] Do not read or program unused areas to prevent the CPU works incorrectly.
- □ [2.6] The contents of the RAM area are undefined at power-on and system reset. Initialize this area by the software.
- □ [2.6] Read the data flash area in bytes.
- □ [2.8.2] If the entire LSI is reset through a system reset, the remapping function is disabled as the REMAPADD register is restored with the initial value.

3. Reset Function

- □ [3.3.1] The BRK instruction reset only initializes the CPU if ELEVEL is 2 or higher. Peripheral circuits and other circuits are not initialized. Use the pin reset or the watchdog timer (WDT) reset to surely initialize the LSI when an abnormality is detected.
- □ [3.3.1] □ Command reset in on-chip debug does not reset to Low Level Detector parts. Do initialization of these functions by writing SFRs on debug, if needed. See Chapter 28 for details .
- □ [3.3.2] In system reset mode, the contents of data memory (RAM) and SFRs that have an undefined initial value are not initialized. Initialize them by the software .
- □ [3.3.4] In case of instantaneous power failure and a pulse shorter than the power-on reset reaction time is asserted to VDD, MCU may not get reset and it may malfunction. In that case, please have preventive measures such as using bypass capacitor to avoid the instantaneous voltage drop or using pin reset to initialize MCU.

4. Power Management

- □ [4.1.3] In order to improve the noise resistance, place the inter-power supply bypass capacitor (CV) and the internal logic voltage (VDDL) capacitor (CL : 1μF) in the vicinity of LSI on the user board using the shortest possible wiring without passing through via holes.
- □ [4.2.2] Writing to the stop code acceptor is invalid on the condition both interrupts enable bits and interrupt request bits are "1", it will not get enabled for entering to the STOP mode.
- □ [4.2.3] The operating state does not enter the standby mode under some conditions. See "4.3.2.3 Note of entering to the standby mode" for detail conditions .
- □ [4.2.3] When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".
- □ [4.2.3] □ Insert two NOP instructions in the next to the instruction of that sets HLT and STP bit to "1". The operation without the two NOP instructions is not guaranteed.
- □ [4.2.6] Do not enter the standby mode when SOFTR bit is "1". Ensure SOFTR bit is "0" before entering the standby mode.
- □ [4.2.9, 4.2.13] DCKACC bit can be set to "0" when the multiplication/division library "muldivu8.lib" is specified. See a manual of the multiplication/division library for how to use.
- □ [4.3.2.3] Note of entering to the standby mode

- □ [4.3.2.4] Note on Return Operation from Standby Mode
- □ [4.3.2.4] Since up to two instructions are executed during the period between the release of standby mode and a transition to interrupt processing, place two NOP instructions next to the instruction set for the standby mode. When a master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", following the execution of the two NOP instructions, the interrupt transition cycle will be executed and execution of the instruction for interrupt routine begins. If MIE is "0", following the execution of the two NOP instructions, the instruction execution is continued from the one that follows the NOP instruction without transition to the interrupt.
- □ [4.3.2.6] Whenthe FHWUPT register is set to "0x01", the frequency of PLL oscillation clock gradually increases and reaches the target frequency switched by the code option before approx. 2 ms elapse. The PLL oscillation clock during this time period can be used for the SYSCLK, however, accuracy of the frequency is not guaranteed.
- □ [4.3.3] If the clock supply is only stopped without resetting each peripheral circuit using the block control function, it may cause the output levels of the timer and communication pins to be fixed, causing the excess current to flow. Also, in the successive approximation type A/D converter, the circuits may stop their function with the current kept flowing.

5. Interrupts

- □ [5.2.3,5.2.5,5.2.7,5.2.9,5.2.16,5.2.20] In the 16-bit timer 0 interrupt, the interrupt factor is assigned to two places. (TM0INT0, TM0INT1)
  Do not use TM0INT and TM0INT1 at the same time. (See Section 5.3.9 "16-bit Timer 0 Interrupt").
- □ [5.2.6 to 5.2.9] When rewrite a specific bit of this register, use a bit symbol. Otherwise, other request bits in the same register might be cleared. See Section 5.3.8 "Writing to IRQ01/IRQ23/IRQ45/IRQ67" for more detail.
- □ [5.2.10] Disable the interrupt level control function by resetting ILE bit to "0" after resetting the Interrupt level control register 0 to 7 (ILC0 to ILC7) to "0x0000" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled (IE01 to IE67 registers are "0x00").
- □ [5.2.10] Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled (IE0 to IE7 registers are "0") or master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.
- □ [5.2.13 to 5.2.20] Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.
- □ [5.3] The WDT interrupt (WDTINT) is a non-maskable interrupt. If the non-maskable interrupt occurs while an interrupt processing is in progress, abort the interrupt processing and proceed with processing the non-maskable interrupt preferentially regardless of multiple interrupts enabled/disabled.
- □ [5.3] For failsafe, define unused all interrupt vectors. If an unused interrupt occurs, it may indicate the possibility that the CPU went out of control. It is recommended to cause the WDT overflow reset to occur using the infinite loop to initialize the LSI.
- □ [5.3.4] Notes on Interrupt Routine (with Interrupt Level Control Disabled)
- □ [5.3.4] Do not enable interrupts in a subroutine called from an interrupt routine for which multiple interrupts are disabled. Otherwise, the program may run out of control when multiple interrupts occur .
- □ [5.3.5] For processing of non-maskable interrupt, follow the flow chart "When multiple interrupts are enabled". Registers that should be saved in the stack are ELR2 and EPSW2.
- □ [5.3.5] When programming in C, it is not required to write program codes for saving/restoring registers because they are generated in the C compiler. However, program codes for enabling/disabling interrupts through EI and DI instructions and for writing to the current interrupt level management register (CIL) must be written. See Section 5.3.6 "How To Write Interrupt Processing When Interrupt Level Control Enabled" for the specific program description.
- □ [5.3.6.1] Do not enable interrupts in a function called from a function for which multiple interrupts are disabled. Otherwise, the program may run out of control when the multiple interrupts occur .

6. Clock Generation Circuit

- □ [6.1.2] After the power-on or the system reset, LSCLK0 (32.768 kHz) is initially selected as SYSCLK.
- □ [6.3.4] While the CPU is running with the low-speed clock, if running the peripheral circuits with the high-speed clock which can frequently generate interrupts, the operation may fail to function properly due to the CPU becoming incapable of processing interrupts in time. If interrupts frequently occur for reasons such as short interrupt cycles of peripheral circuits, take into account the operating frequency of the CPU so that it can process interrupts in time.

8. 16-bit Timer

- □ [8.2.2] When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD.
- □ [8.2.3] Read TMHnC register twice to verify the valid data to prevent reading uncertain data while counting-up, if a source of timer clock is as different as one of system clock.
- □ [8.2.4] Set TMHnMOD when the timer n is stopped (THnSTAT bits of TMHSTAT/TMHXSTAT register are "0"). If it is changed while it is operating, the operation is not guaranteed.
- □ [8.3.2] After THnRUN bit is set to "1", the first interrupt has a time error equivalent to maximum of one clock of the timer clock because the counting operation starts in synchronization with the timer clock. The 2nd timer interrupt or later interrupts have constant cycles.

□ [8.3.2] After THnSTP bit is set to "1", a 16-bit timer n interrupt (TMnINT) may be generated depending on the stop timing because the counting operation stops in synchronization with the timer clock .

9. Operational Timer

□ [9.2.2] When 0x0000 is written in this register, 0x0001 is set and the read value is also becoming 0x0001.
□ [9.2.8] When switching from PWM mode to CAPTURE mode, OTnFLGA bit and OTnFLGB bits might be unintentionally set to "1". So, when switching to CAPTURE mode, read OTn0R register and OTn0F register once to initialize OTnFLGA and OTnFLGB bits.
□ [9.2.9] Set OTnMOD register when the counter operation is stopped.
□ [9.2.10] If it is required to update OTnOCON register while the counter is running, change the setting value after stopping the logical output by OTC0OCON/OTC1OCON register. If the setting of OTnOCON register is changed during counter operation in the logical output permitting state, unintended waveforms might be output.
□ [9.2.11] Set OTnSCLR register when counter operation is stopped.
□ [9.2.11] The time interval set to OTnDLY bit must be shorter than the interval set to OTnP register.
If the time interval is set for OTnDLY bit longer than the interval set to OTnP register, the counter may stop even if the time interval is less than the interval set by OTnDLY bit
[9.2.12] Set OTnCLK register when counter operation is stopped.
□ [9.2.13,9.2.14] Trigger pulse width must be 2cloks wide of timer clock or more.
□ [9.2.13,9.2,14] In one-shot mode, set OTnTRM02 to OTnTRM00 bits to "1X0" or "1X1", and if the start condition and the level of the counter control signal 0 match, and the stop condition in one-shot mode is satisfied, the counting operation is continued (count-up is resumed from 0).
□ [9.2.13,9.2.14] If OTnCTRG0 register is set when a trigger event is enabled, a trigger event may occur at the moment of the setting. So when setting OTnCTRG0 register, verify that OTnTGEN bit of OTCSTAT register is "0" and change the setting of OTnCTRG0 register in the trigger operation stopped state.
□ [9.2.15] When changing the setting of OTnFTRG0 register during counter operation, set the following bits of the register to "0" once, and then change the setting value.
□ ・OTnFCC0 bit
□ ・OTnFCN01 bit
□ ・OTnFCN00 bit
□ [9.2.16]
□ When changing the setting of OTnFTRG1 register during counter operation, set the following bits of the register to "0" once, and then change the setting value.
□ ・OTnFCC1 bit
□ ・OTnFCN11 bit
□ ・OTnFCN10 bit
□ [9.2.28] Set OTCPC register in a state where the counter operation is stopped.
□ [9.2.28] When the period is set to be linked with OT20R/OT30R, set the period of the count clock to be set to the linked OTMn so that it is less than or equal to the period of the count clock set in OTM2 and OTM3.
□ [9.2.28] If the period is set to longer than the period of the count clock set in OTM2 and OTM3, there is a possibility that an unintended value may be set to OTnP during period-linked operation.

10. Watchdog Timer

□ [10.1.1] Not all the abnormal operations are detectable by the watchdog timer. Even if the CPU is runaway , the watchdog timer is undetectable to the abnormality in the operation state in which the WDT counter is cleared. As a fail-safe, it is recommended that the WDT counter is cleared at one place in the main loop of the program.
□ [10.2.2] In WDT interrupt routine (when the interrupt level (ELEVEL) of the CPU program status word (PSW) is "2"), WDT counter is unable to get cleared.
□ [10.2.3] See the data-sheet for frequency accuracy of RC1K.
□ [10.3.1] In the STOP mode, the WDT timer is stopped.
□ [10.3.3] When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.
□ [10.3.3] In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.

13. I^2C Bus

□ [13.1.3] Use external pull-up resistors for SDA pin and SCL pin according to the I2C bus specification. The internal pull-up resistors are unsatisfied the I2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.
□ [13.1.3, 13.2.2] If powering off this LSI in the slave mode, it disables communications of other devices on the I2C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.

- □ [13.1.4, 13.2.2] Do not connect multiple master devices on the I2C bus when using the master function.
- □ [13.2.2] All SFRs are shared in master mode with slave mode. If switching master/slave mode, set "0" to I2U0EN bit of I2UMOD register, then switch the mode and reconfigurate each SFRs.
- □ [13.3.4] Update it without a bit access instruction in the control register setting wait state. See Section 13.5.1.5 "Control Register Setting Wait State" for details.
- □ [13.3.4] When I2U0ST bit is "1", write other bits of I2U0CON register in the control register setting wait state .
- □ [13.4.4] If system clock is extremely slower than the communication speed, the data transmission/reception can be failed .
- □ [13.4.4] Before releasing the communication wait state, change the system clock enough speed for the communication.
- □ [13.4.5] To be disable the wake-up from standby mode by matching the slave address, Stop the operation by resetting I2U0EN bit to "0" before entering STOP mode.
- □ [13.5.4] When the slave device uses the clock stretch function which holds SCLU0 pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.
- □ [13.6.2] If entering to the STOP mode while the slave mode is enabled, first make sure that communication is not in progress (from coincidence of address to reception of stop condition).

14.UART

- □ [14.2.6] Do setting for used ports and the mode/baud rate before setting "1" to UnEN bit.
- □ [14.2.7] Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0).
- □ [14.2.9, 14.2.10] Be sure to set the UAnBRT and UAnBRC register while communication is stopped (UnEN=0). Do not rewrite it during communication.
- □ [14.3.4] The transmission is start when setting "1" to UnEN bit of UAnCON with the UnFUL bit =1. Write "1" to the UnFULC bit in the UAnSTAC register to reset the UnFUL bit, and then set "1" to the UnEN to allow transmission/reception, if the transmission data is not ready and the reception is permitted first.
- □ [14.3.5.3] When designing the system, consider the difference of the baud rate between the transmission side and reception side, a delay of the start bit detection, signal degradation and noise influence, then adjust the baud rate and reception timing to ensure sufficient receiving margin.

17. GPIO

- □ [17.2.3] P00 pin is initially configured as the input with pull-up resistor. If input "L" level at an initial setting, the input current flows.
- □ [17.2.3 ~ 17.2.7] Be sure to set PnMODm (m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting PnMOD01 register when the interrupt is enabled, unexpected interrupts may happen.
- □ [17.2.3 ~ 17.2.7] It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.
- □ [17.2.3 ~ 17.2.7] Don't set un-assigned shared functions on PnmMD3-0 bits.
- □ [17.3.8] Notes for using P00/TEST0 pin

18. External Interrupt Function

- □ [18.2.3] If high-speed clock is selected as sampling clock source, it works without sampling when the high-speed clock does not supply; it includes stop by entry to standby mode. Set to LSCLK0 as sampling clock if needed..
- □ [18.2.3] In the STOP mode, it works without sampling.

20. Analog comparator

- □ [20.1.3] When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers to set the general port to Hi-impedance, otherwise a shoot-through current may flow.
- □ [20.1.3] An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.
- □ [20.2.2] When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers ()n=0 to 2, m=0 to 7) to set the general port to Hi-impedance, otherwise a shoot-through current may flow.
- □ [20.2.2] An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.
- □ [20.2.2] After CMPnEN bit is set to "1", it takes 100μs to stabilize the operation of the analog comparator.
- □ [20.2.3] In the STOP mode, the sampling clock stops and the VLS works without sampling regardless the setting in CMPnCS1 bit and CMPnCS0 bit. When selecting "with sampling" and entering those mode, there is a time period in which interrupts gets disabled.
- □ [20.2.3] When the HSCLK is selected for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work. When using analog comparator in this case, select "No sampling" or "Sampling with LSCLK0" for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".
- □ [20.2.3] Write CMPnMOD register when the Analog comparator stops (CMPnEN bit of CMPCON register is "0"), The analog comparator output is not guaranteed CMPnMOD register is changed while the analog comparator is activated.
- □ [20.2.3] When CMPnOMASK bit is set to "1", set to interrupt inhibition at CMPnE1/CMPnE0 bit. If interrupts are enabled, interrupts may occur unintentionally during mask processing at the start of A/D conversion.

□ [20.2.3] When the output of the D/A converter is selected as CMPnM pin and the DAC internal reference voltage (0.8V) is selected as the reference voltage of the D/A converter (DACMODE.DAMDn bit = "1"), enable sampling function of the analog comparator.
□ [20.2.4] After the analog comparator is started, CMPnSTA bit may be set to "1" until the operation stabilizes. After the operation is stable, read CMPnSTA register once and initialize the value.

21. D/A Converter

□ [21.1.3, 21.2.2] For GPIO that outputs D/A converter, set the PnmMD2~0 bits (m: bit number 0~7) of the corresponding port n-mode register m to the 5th function, write "0" to the PnmIE bit and the PnmOE bit, and set it to high impedance (input disabled, output disabled). In other settings, the D/A converter is not output.

22. LOW Level Detector

□ [22.2.2, 22.2.3] When the LLD is in enable operating (LLD0EN=1) and the STOP mode is shifted, make sure that the LLD is in stable operation (LLD0RF bit = 1) before transitioning.
□ [22.2.2] If a reset other than Power On Reset and pin reset occurs during LLD operation, the LLD maintains its operating state.
□ [22.2.4] During STOP, this register is set "no sampling" regardless of LLD0SM1 bit setting.
□ [22.3.1.1] It is im possible to change to STOP mode during the LLD stabilization time. When changing to STOP mode after LLD starting, make sure that LLD0RF bit is set to "1" before changing STOP mode.

23. Successive Approximation Type A/D Converter

□ [23.1.3] When using the SA-ADC, set PnmMD2 to 0 bit to the 6th or 8th function of port n mode register 01/23/45/67 (n: port number 1, 2 m: bit number 0 to 7) and set PnmIE bit and PnmOE bit to "0" as "Disable input" and "Disable output", otherwise a shoot-through current may flow.
□ [23.1.3] While the A/D converter is operating, an influence of the noise is reducible by preventing the switching of neighboring pins or A/D converting in the HALT mode.
□ [23.1.3] When using AIN4, if both P13 and P20 are set to the 6th or 8th function, the input from P13 becomes AIN4 input. Input from P20 will be disabled.
□ [23.2.3] Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN0 and SADEN1). If no channel is chosen, the operation does not start.
□ [23.2.3] Enter STOP mode after checking SARUN bit is "0". It does not enter the STOP mode when SARUN bit is "1".
□ [23.2.3] When SACK2 to 0 bits are set to 0x7, it takes max. 3 clocks of the low-speed clock (LSCLK0) to start or stop the A/D conversion after setting or resetting SARUN bit.
□ [23.2.4] If only temperature conversion is enabled and the SAD_CLK is set to 1/1 x HSCLK, the SAD_CLK period used for the conversion interval for A/D conversion will be 1/2 x HSCLK.
□ [23.2.5] After VREFEN bit is set to "1", it takes 200μs (Max.) for the ADC internal reference voltage (2.5 V) to stabilize. A/D conversion should be started after this stabilization time has elapsed. VREFEN bit is set to "1" and remains stable after 200μs until it is disabled. Therefore, there is no need to wait for 200μs (Max.) again when A/D conversion.
□ [23.2.8, 23.2.9] Do not start the A/D conversion when the all bits of SACHn (n=00 to 18) are "0". In that case SARUN bit of SADCON register does not get to "1".
□ [23.2.9] When using channel 16 (SACH16), permit the temperature sensor operation by VREFCON register.
□ [23.2.9]When using channel 18 (SACH18), enable the A/D converter test function by SADTMOD register (set to something other than "00").
□ [23.2.10] When the setting of the source clock of the SAD_CLK and the setting of the system clock are different, the wrong value may be read during the update. Make sure that SARUN bit of SADCON register is "0" before reading. Also, during continuous A/D conversion (SALP=1), read SADR register twice and make sure that the values are the same.
□ [23.2.11] If the source clock (HSCLK/LSCLK0) setting of the SAD_CLK differs from the source clock (HSCLK/LSCLK0) setting of the system clock, uncertain data during the A/D conversion result update may be read when the SADRn register is read. If the source clock settings of the SAD_CLK and the system clock are different, make sure that the SARUN bit of the SADCON register is "0" before reading. During continuous A/D conversion (the SADP bit of the SADMOD register is "1"), read the SADR register twice and make sure that the values are the same.
□ [23.4] Notes on SA-ADC

24. PGA

□ [24.1.3, 24.2.2] When PGA is used, following the setting below.
(1) Set PnmMD2 to 0bit of the corresponding port n-mode register m (n: 0 to 2, m: 0 to 7) to the 6th or 8th function,
(2) Write "0" to the corresponding bits of input permission (PnmIE) and output permission (PnmOE).
(3) set the general-purpose port to high impedance.
In other settings, through current may occurs.
□ [24.1.3] During PGA is in operation, the influence of the output voltage fluctuates due to the noise can be reduced by not switching adjacent pins on AINn.

- □ [24.2.2] After setting the PGAEN bit to "1", it takes 200µs for PGA to stabilize. A/D conversion should be started after this stabilization time has elapsed.
- □ After the stabilization time (200µs) has elapsed, PGA remains stable until the operation stops, so there is no need to wait for 200µs again for each A/D conversions.
- □ [24.2.3] Writing to the PGAMOD register while PGA operation is permited, the operation is not guaranteed.
- □

26. Flash Memory

- □ [26.2.2] Note that programming for the program memory space is performed by the unit of 2 bytes. Because of this, the setting values in the FA0 bit are ignored.
- □ [26.2.4] Specify a segment address to FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.
- □ [26.2.4] During programming data-flash, a CPU can execute instruction by the background operation function; BGO. Confirm FDPRSTA bit of FLASHSTA register for complition of programming.
- □ [26.2.4] Erase data in the addresses in advance. Programmed data without erase is unguaranteed.
- □ [26.2.4] Do not read or program unused areas to prevent the CPU works incorrectly.
- □ [26.2.6, 26.2.7] A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.
- □ [26.2.8] Perform the erasing or programming after checking FDERSTA bit or FDPRSTA bit are "0". Do not perform the erasing or programming when either FDERSTA bit or FDPRSTA bit is "1".
- □ [26.3.1] Notes on Debugging Self-programming Code
- □ [26.3.2] Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.
- □ [26.3.2] During block/sector erasing, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
- □ [26.3.2, 26.3.3] For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of FLASHCON register to "1".
- □ [26.3.2] Only write areas irrelevant to program processing. If programming the area where program processing is in progress, the LSI works incorrectly.
- □ [26.3.2] During the programming, the CPU stops the operation for maximum 40 µs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
- □ [26.3.2] For data programming setting, place two NOP instructions following the instruction used to set the programming data in FLASHD0 register.
- □ [26.3.3] STOP mode is not available during the erasing. In addition, set FSELF bit of FLASHSLF register to "0" after the erasing is completed.
- □ [26.3.3] The data flash area is unreadable during erasing.
- □ [26.3.3] The CPU continues program processing even while data flash programming is in progress. An entering to the STOP mode is not available during the programming. In addition, set FSELF bit of FLASHSLF register to "0" (erase/program disabled) after the programming ended.
- □ [26.3.3] The data flash area is unreadable during programming.
- □ [26.3.3] For data programming setting, place two NOP instructions following the instruction used to set the programming data in FLASHD0L register.
- □ [26.3.4] Notes on use of self-programming
- □ [26.4.3] Accessing to the program code area is performed in units of four bytes. Set four byte boundaries (0H/4H/8H/CH) for lower four bits of the address. Accessing to the data flash area is performed in units of one byte.
- □ [26.4.4] Perform the process within 55 ms from the ISP mode transition command (B in Figure 26-6) to the completion of the initial setting command transmission (1).
- □ [26.4.5.1] Initial setting process should be processed within 1000 ms.
- □ [26.4.5.2] Erasure should be processed within 500 ms.
- □ [26.4.5.3] Writing should be processed within 500ms. When writing multiple addresses, process from data setting to data setting of the next address or completion of sending initial setting command (7) within 500 ms.
- □ [26.4.5.4] Verify should be processed within 500 ms. When verifying multiple addresses, process from data setting to data setting of the next address or completion of sending initial setting command (7) within 500 ms.

28. On-Chip Debug Function

- □ [28.3] Make RESET_N pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.
- □ [28.3] Make P00/TEST0 pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.
- □ [28.3] Do not program instruction codes into the LSI that set P00/TEST0 pin to the output mode. If P00/TEST0 is set to the output mode before On-chip emulator performs read/write to/from the target chip, communication with On-chip emulator after that will be disabled. Also note that the input/output mode of P00/TEST0 is uninitialized by On-chip emulator.

- □ [28.3] Validate the ROM code on user production board without the On-chip emulator.
- □ [28.3] Disconnect On-chip emulator when measuring the current consumption of the target system. If On-chip emulator remains connected, the current consumption increases as the on-chip debug circuit inside the LSI works for the communication.
- □ [28.3] LSI used to debug a program is not covered by the product warranty. Do not use the LSI for mass-production.
- □ [28.3] A reset due to unused ROM area access does not occur in the on-chip debug mode regardless of code option settings.
- □ [28.3] A RAM parity error reset does not occur in the on-chip debug mode and the break operation occurs instead.
- □ [28.3] If the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including when the RAM is not initialized), a RAM parity error may occur even if the RAM area is not displayed.
- □ [28.3] The all interrupts and watchdog timer operation always stop while the debugger is in the break state.
- □ [28.3] If the LSI cannot be connected to the debugger, check that the LSI is supplied with a power supply (V_{DD}=4.5V~5.5V).

29. Safety Function

- □ [29.2.5] If the RAM is not initialized when it is read, a parity error may occur. If you want to use parity error reset (check the parity error flag or enable parity error reset), initialize the entire RAM area before using it.
- □ [29.2.7] If the RAM is not initialized when it is read, a parity error may occur.
- □ [29.2.8] If the MCISTATL register is not zero, a request to interrupt controller is not given when a new interrupt occurs. Clear the MCISTATL register with the MCINTCL register before that time.

30. Code Option

- □ [30.2.1] There are available to read the code option values from SFRs, if INITE flag bit of Reset Status Register (RSTAT) is "0".
- □ [30.3] For the code option data definition, always use the dw directive instruction to configure the data in the unit of word.
- □ [30.4] At the time of shipment or when the program is erased, the code options are also initialized and the branch trace function is disabled. If you want to enable branch trace function for an LSI that is in disabled branch trace function state, connect to a debugger with the normal product name and enable branch trace function.

31. Auxiliary Function

A. SFR List

- □ Access "Reserved" register is not guaranteed. Please do not access them.

B. Package Dimensions

- □ The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).
- □ The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

C. Instruction Execution Cycle

D. Application Circuit Example

- □ Place the capacitor for VDDL pin as close to the LSI power pins as possible.

ROHM Co.,Ltd.

Revision History

REVISON HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q2045-01	2024.6.21	—	—	1st edition
FEUL62Q2045-02	2024.9.5	1-10	1-10	1.3.2 Correction of 'Pin List'.
		22-6	22-6	22.3 Correction of 'Description of Operation'.
		22-7, 22-8	22-7, 22-8	Figure 22-3, 22-4 Added symbol descriptions