Dear customer

ROHM Co., Ltd. ("ROHM"), on the 1st day of April, 2024,
has absorbed into merger with 100%-owned subsidiary of LAPIS Technology Co., Ltd.

Therefore, all references to "LAPIS Technology Co., Ltd.", "LAPIS Technology"
and/or "LAPIS" in this document shall be replaced with "ROHM Co., Ltd."
Furthermore, there are no changes to the documents relating to our products other than
the company name, the company trademark, logo, etc.

Thank you for your understanding.

ROHM Co., Ltd.
April 1, 2024

Dear customer

LAPIS Semiconductor Co., Ltd. ("LAPIS Semiconductor"), on the 1st day of October, 2020, implemented the incorporation-type company split (shinsetsu-bunkatsu) in which LAPIS established a new company, LAPIS Technology Co., Ltd. (“LAPIS Technology”) and LAPIS Technology succeeded LAPIS Semiconductor’s LSI business.

Therefore, all references to "LAPIS Semiconductor Co., Ltd.", "LAPIS Semiconductor" and/or "LAPIS" in this document shall be replaced with "LAPIS Technology Co., Ltd."

Furthermore, there are no changes to the documents relating to our products other than the company name, the company trademark, logo, etc.

Thank you for your understanding.

LAPIS Technology Co., Ltd.
October 1, 2020

ROHM GROUP
LAPIS
SEMICONDUCTOR

FEUL62Q1200A-06

ML62Q1200A Group
User's Manual

Issue Date: Feb 8, 2019

ROHM
SEMICONDUCTOR

Notes

1) The information contained herein is subject to change without notice.

2) Although LAPIS Semiconductor is continuously working to improve product reliability and quality, semiconductors can break down and malfunction due to various factors. Therefore, in order to prevent personal injury or fire arising from failure, please take safety measures such as complying with the derating characteristics, implementing redundant and fire prevention designs, and utilizing backups and fail-safe procedures. LAPIS Semiconductor shall have no responsibility for any damages arising out of the use of our Products beyond the rating specified by LAPIS Semiconductor.

3) Examples of application circuits, circuit constants and any other information contained herein are provided only to illustrate the standard usage and operations of the Products.The peripheral conditions must be taken into account when designing circuits for mass production.

4) The technical information specified herein is intended only to show the typical functions of the Products and examples of application circuits for the Products. No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of LAPIS Semiconductor or any third party with respect to the information contained in this document; therefore LAPIS Semiconductor 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) The Products are intended for use in general electronic equipment (i.e. AV/OA devices, communication, consumer systems, gaming/entertainment sets) as well as the applications indicated in this document.

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

7) For use of our Products in applications requiring a high degree of reliability (as exemplified below), please contact and consult with a LAPIS Semiconductor representative: transportation equipment (i.e. cars, ships, trains), primary communication equipment, traffic lights, fire/crime prevention, safety equipment, medical systems, servers, solar cells, and power transmission systems.

8) Do not use our Products in applications requiring extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters.

9) LAPIS Semiconductor shall have no responsibility for any damages or injury arising from non-compliance with the recommended usage conditions and specifications contained herein.

10) LAPIS Semiconductor has used reasonable care to ensure the accuracy of the information contained in this document. However, LAPIS Semiconductor does not warrant that such information is error-free and LAPIS Semiconductor shall have no responsibility for any damages arising from any inaccuracy or misprint of such information.

11) Please use the Products in accordance with any applicable environmental laws and regulations, such as the RoHS Directive. For more details, including RoHS compatibility, please contact a ROHM sales office. LAPIS Semiconductor shall have no responsibility for any damages or losses resulting non-compliance with any applicable laws or regulations.

12) 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.

13) This document, in part or in whole, may not be reprinted or reproduced without prior consent of LAPIS Semiconductor.

LAPIS Semiconductor Co.,Ltd.

2-4-8 Shinyokohama, Kouhoku-ku,
Yokohama 222-8575, Japan
http://www.lapis-semi.com/en/

Preface

This manual describes the operation of the hardware of the 16-bit microcontroller ML62Q1200A Group.

The following manuals are also available. Read them as necessary.

■ nX-U16/100 Core Instruction Manual
Description on the basic architecture and the each instruction of the nX-U16/100 Core.

■ MACU8 Assembler Package User's Manual
Description on the method of operating the relocatable assembler, the linker, the librarian, and the object converter and also on the specifications of the assembler language.

■ CCU8 User's Manual
Description on the method of operating the compiler.

■ CCU8 Programming Guide
Description on the method of programming.

■ CCU8 Language Reference
Description on the language specifications.

■ DTU8 Debugger User's Manual
Description on the method of operating the debugger DTU8.

■ IDEU8 User's Manual
Description on the integrated development environment IDEU8.

■ EASE1000 User's Manual
Description on the on-chip debug emulator EASE1000.

■ MWuEASE Flash Writer Host Program User's Manual
Description on the Flash Writer host program.

Notation

Classification	Notation	Description
♦ Numeric value	xxh, xxH	Indicates a hexadecimal number. x: Any value in the range of 0 to F
	xxb	Indicates a binary number; "b" may be omitted. x: A value 0 or 1
♦ 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 voltage signal levels V_{IH} and V_{OH} as specified by the electrical characteristics.
	"L" level	Indicates low voltage signal levels V_{IL} and V_{OL} as specified by the electrical characteristics.

♦ 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.
"-" means an invalid bit. This bit returns "0" for read and write to this bit is ignored, unless otherwise specified.

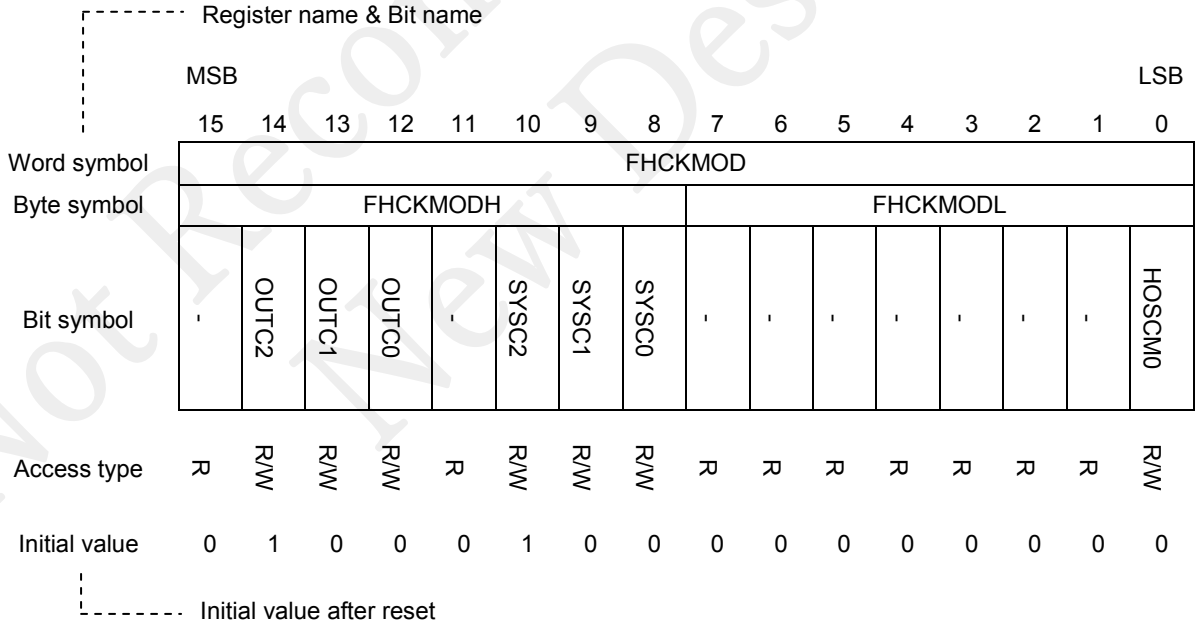


Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 13

Chapter 21

Chapter 22

Chapter 23

Chapter 24

Chapter 25

Chapter 26

Chapter 27

Chapter 28

Chapter 29

Appendixes

Revision History

Chapter 1 Overview

1.Overview

1.1 Features

ML62Q1200A Group is a high performance CMOS 16-bit microcontroller equipped with an 16-bit CPU nX-U16/100 and integrated with program memory(Flash memory), data memory(RAM), data Flash and rich peripheral functions such as the multiplier/divider, CRC operator, DMA controller, clock generator, timer, UART, synchronous serial port, I^2C bus interface unit, buzzer, Voltage Level Supervisor(VSL), successive approximation type A/D converter, D/A converter , analog comparator, safety function and etc.
The CPU nX-U16/100 is capable of efficient instruction execution in 1-intruction 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.
The ML620Q1200A Group has five packages (16pin - 32pin) and five kinds of memory sizes(16Kbyte – 64Kbyte).

Table 1-1 ML62Q1200A Group Product List

Program memory	Data memory (RAM)	Data Flash	16pin SSOP16 WQFN16	20pin TSSOP20	24pin WQFN24	32pin TQFP32
64Kbyte	4Kbyte	2Kbyte	–	–	ML62Q1247A	ML62Q1267A
48Kbyte	4Kbyte	2Kbyte	–	–	ML62Q1246A	ML62Q1266A
32Kbyte	4Kbyte	2Kbyte	–	–	ML62Q1245A	ML62Q1265A
	2Kbyte	2Kbyte	ML62Q1225A	ML62Q1235A	–	–
24Kbyte	2Kbyte	2Kbyte	ML62Q1224A	ML62Q1234A	–	–
16Kbyte	2Kbyte	2Kbyte	ML62Q1223A	ML62Q1233A	–	–

- CPU
  - 16-bit RISC CPU (CPU name: nX-U16/100)
  - Instruction system: 16-bit length instruction
  - 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
  - On-chip debug function built-in (supported by LAPIS on-chip debug emulator EASE1000)
  - ISP (In-System Programming) function built-in
  - Minimum instruction execution time
    30.5 μs (at 32.768 KHz system clock)
    62.5ns/41.6ns (at 16 MHz/24MHz system clock)

- Coprocessor for multiplication and division
  - Multiplication: 16bit × 16bit (operation time 4 cycles)
  - Division: 32bit / 16bit (operation time 8 cycles)
  - Division: 32bit / 32bit (operation time 16 cycles)
  - Multiply-accumulate (non-saturating): 16bit × 16bit + 32bit (operation time 4 cycles)
  - Multiply-accumulate (saturating): 16bit × 16bit + 32bit (operation time 4 cycles)

- Operating voltage and temperature
  - Operating voltage: VDD = 1.6 to 5.5 V
  - Operating temperature: -40 to 105 °C

- Internal memory
  - Prrogram Flash memory area
    Rewrite count: 100 cycles
    Rewrite unit: 32bit(4byte)
    Erase unit: 16Kbyte/1Kbyte
    Erase/Rewrite temperature: 0°C to +40°C

  - Data Flash memory area
    Rewrite count 10,000 cycles
    Rewrite unit: 8bit(1byte)
    Erase unit: 2Kbyte/128byte
    Erase/Rewrite temperature: -40°C to +85°C
    Back Ground Operation(CPU can work while erasing and rewriting)

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

- Clock
  - Low-speed clock
    Internal low-speed RC oscillation (32.768 KHz)
  - High-speed clock
    PLL oscillation (32MHz/24MHz/16MHz is selectable by flash code option)
  - WDT(Watch Dog Timer) independent clock
    Internal low-speed RC oscillation (1kHz)

- Reset
  - RESET_N pin reset
  - Reset by power-on detection
  - Reset by the 2nd watchdog timer (WDT) overflow
  - Reset by counter clear during the windows close of watchdog timer (WDT)
  - Reset by RAM parity error
  - Reset by voltage level detection (VLS)
  - Reset by invalid memory access (detecting abnormal program counter)
  - The software reset by BRK instruction (reset CPU only)

- Power management
  - HALT mode: CPU stops executing instruction, clock oscillations and peripheral circuits remain previous states
  - HALT-H mode: CPU stops executing instruction, high-speed clock oscillation stops and peripheral circuits working with low-speed clock remain previous states
  - STOP mode: CPU stops executing instruction, both high-speed oscillation and low-speed oscillation stop.
  - STOP-D mode: CPU stops executing instruction, both high-speed oscillation and low-speed oscillation stop. The internal regulator's output voltage (V_{DDL}) goes down to reduce the current consumption (RAM data are retained).
  - Clock gear: The frequency of high-speed system clock can be changed by software (1/1, 1/2, 1/4, 1/8, 1/16 or 1/32 of the oscillation clock)
  - Block Control Function: Powers down the circuits of unused function blocks (reset the block or stop supplying the clock)

- Interrupt controller
  - Non-maskable interrupt source: 1 (Internal sources: WDT)
  - Maskable interrupt sources: max.31 (ML62Q126xA/32pin: Internal sources: max.23, External sources: 8)
  - Four step interrupt levels

- Watchdog timer(WDT)
  - Operation clock: 1kHz WDT independent clock or 32.768kHz RC oscillation clock, selectable by code option
  - Overflow period: 8 types selectable (8ms, 16ms, 32ms, 64ms, 125ms, 500ms, 2000ms and 8000ms @32.768kHz)
  - WDT counter clear enable period : 50%, 75% or 100% of overflow period
    When 100% of overflow period is selected,
    The first overflow generates an interrupt, and the second overflow generates a reset.
    When 50% or 70% of overflow period is selected,
    The first overflow generates a reset.
    Clearing the WDT counter out of the enable period generates the WDT invalid clear reset.
  - WDT operation : Enable or disable is selectable by code option
  - Readable WDT counter (WDT counter monitor function)

- DMA(Direct Memory Access) controller
  - Operation mode : Wait mode
  - Channel : 2ch
  - Transfer unit: 8bit/16bit
  - Max. transfer count: 1024 time
  - Transfer type: 2 cycle transfer
  - Transfer mode: Single transfer mode
    Fixed address, address increments and address decrements
  - Transfer target: SFR/RAM ←→ SFR/RAM (Transfer from/to Flash is not supported)
  - Transfer request: Serial unit interrupt, A/D interrupt and Timer interrupt

- Time base counter
  - Devide the Low-speed clock(LSCLK) and generate 32.768kHz~1Hz internal pulse signals
  - Priodical interrupt × 3 selectable from 8 frequencies (128Hz, 64Hz, 32Hz, 16Hz, 8Hz, 4Hz, 2Hz and 1Hz)
  - The time base clock output (1Hz or 2Hz) from general purpose ports (TBCOUT1).

- Functional timer(FTM)
  - Channel: 4ch
  - Timer one shot mode and repeat mode, Caputure mode, PWM mode1 and PWM mode 2(complementary output)
  - Same start/stop is avaible with different channels
    (This function is not avaible with 16bit Genral Timer)
  - Event trigger (external interrupts, analog comprator interrupts, 16bit genral timer interrupts and Functional timer interrupts)
  - Delay counter (for generating dead time for motor control)
  - Available to specify devision ratio of counter clock channel by channel

- 16bit General timers
  - Channel: 6ch
  - 8 bits timer mode and 16-bit timer mode (1ch 16-bit timer is configurable as 2ch 8-bit timer)
  - Same start/stop is avaible with different channels
    (This function is not avaible with Functional Timer)
  - Timer output (toggled by overflow)
  - Available to specify devision ratio of counter clock channel by channel

- Serial communication unit
  - Channel: Max. 2ch
  - Synchronous Serial Port or UART is seletable in each channel

  < Synchronous Serial Port >
  - Master/slave selectable
  - LSB first/MSB first selectable
  - 8-bit length/16-bit length selectable

  < UART >
  - Full-duplex communication x 2 ch(One Full-duplexUART is configurable as two half-duplex UARTs)
  - Bit length, parity/no parity, odd parity/even parity, 1 stop bit/2 stop bits
  - Positive logic/negative logic selectable
  - LSB first/MSB first selectable
  - Internal baud rate generator (1bps ~ 2Mbps)

- I^2C bus interface unit (Master/Slave)
  - Channel: 1ch
  - Master or Slave mode is selectable

  < Master function >
  - Standard mode (100 kbit/s), fast mode (400 kbit/s) and 1Mbps mode(1Mbit/s)
  - Handshake (Clock syncronization)
  - 7bit address format (10bit address format is supported)

  < Slave function >
  - Standard mode (100 kbit/s), fast mode (400 kbit/s) and 1Mbps mode(1Mbit/s)
  - Handshake (Clock syncronization)
  - 7bit address format (10bit address format is supported)

- I^2C bus interface (Master only)
  - Standard mode (100 kbit/s), fast mode (400 kbit/s) and 1Mbps mode(1Mbit/s)
  - Handshake (Clock syncronization)
  - 7bit address format (10bit address format is supported)

- General-purpose ports（including secondary functions）
  - I/O port: Max. 28 (32pinTQFP, including one pin for on-chip debug)
  - External interrput function × 8
  - LED driver port : Max. 27 (32pinTQFP)
  - Carrier frequency output function (used for IR communication)

- Successive approximation type A/D converter
  - Channel: Max.8ch (20pinTSSOP, 24pinWQFN and 32pinTQFP)
  - Resolution: 10bit
  - Conversion time: Selectable 2.25μs (min) /channel (When the conversion clock is 8MHz)$^{)}$
  - VDD, Internal reference voltage(Approx. 1.55V) / Extenal reference voltage (V_{REF} pin) is selectable
  - Scan function (repeat conversion)
  - One result register for each channel
  - Interrupt by threshold of conversion result
  - Temprature sensor for the Low-speed RC oscillation frequency adjustment

- Voltage level superviosr (VLS)
  - Accuracy: ±4%
  - Threshold voltage: 12 values selectable (1.85V ~ 4.00V)
  - Voltage level detection reset (VLS reset)
  - Voltage level detection interrupt (VLS0 interrupt)

- Analog comparator
  - Channel: 1ch
  - Interrupts allow edge selection and sampling selection
  - An external or an internal reference voltage is selectable

- D/A converter
  - Channel: 1ch
  - Resolution: 8bit
  - Output impedance: 6k ohm(Typ.)
  - R-2R radder method

- Buzzer
  - 4 buzzer mode (Repeat sound, Single sound, Intermittent sound 1 and Intermittent sound 2)
  - 8frequencies (4.096kHz to 293Hz)
  - 15 step duty (1/16 to 15/16)
  - Slectable the logic of buzzer output pin (Possitive or Negative logic)

- CRC(Cycle Redundancy Check) operation function
  - Generation eqution: $X^{16}+X^{12}+X^5+1$
  - LSB first
  - Automatic CRC mode: Automatic CRC calculation with data of program memory in HALT mode

- Safety Function
  - RAM/SFR guard
  - Automatic CRC calculation with data of program memory
  - RAM parity error detection
  - ROM unused area access reset
  - Clock mutual check
  - WDT counter check
  - Successive approximation type A/D converter test
  - UART test
  - Synchronous serial test
  - I^2C test
  - GPIO test

- Shipping pacakge

  - 16-pin plastic SSOP
    ML62Q1223A/1224A/1225A-xxxMB (Blank part: ML62Q1223A/1224A/1225A-NNNMB)

  - 16-pin plastic WQFN
    ML62Q1223A/1224A/1225A-xxxGD (Blank part: ML62Q1223A/1224A/1225A-NNNGD)

  - 20-pin plastic TSSOP
    ML62Q1233A/1234A/1235A-xxxTD (Blank part: ML62Q1233A/1234A/1235A-NNNTD)

  - 24-pin plastic WQFN
    ML62Q1245A/1246A/1247A-xxxGD (Blank part: ML62Q1245A/1246A/1247A-NNNGD)

  - 32-pin plastic TQFP
    ML62Q1265A/1266A/1267A-xxxTB (Blank part: ML62Q1265A/1266A/1267A-NNNTB)

    xxx: ROM code number

- ML62Q1200A Group how to read the part number

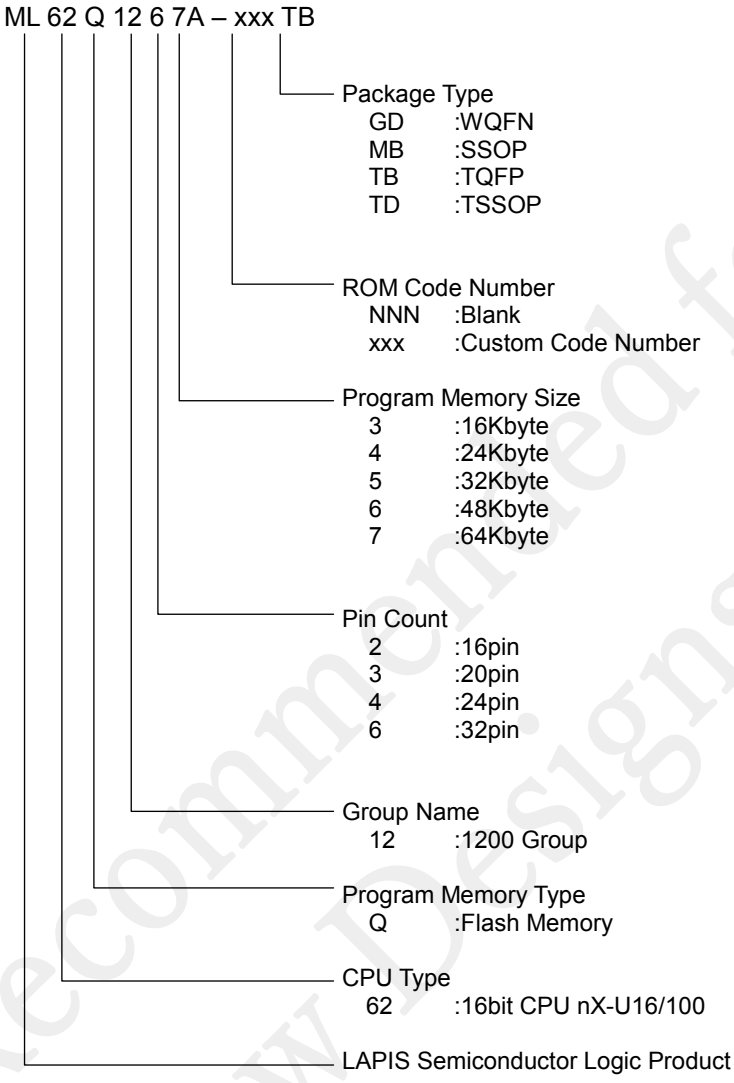


Figure 1-1 ML62Q1200A Group Part Number

- ML62Q1200A Group Main Function List

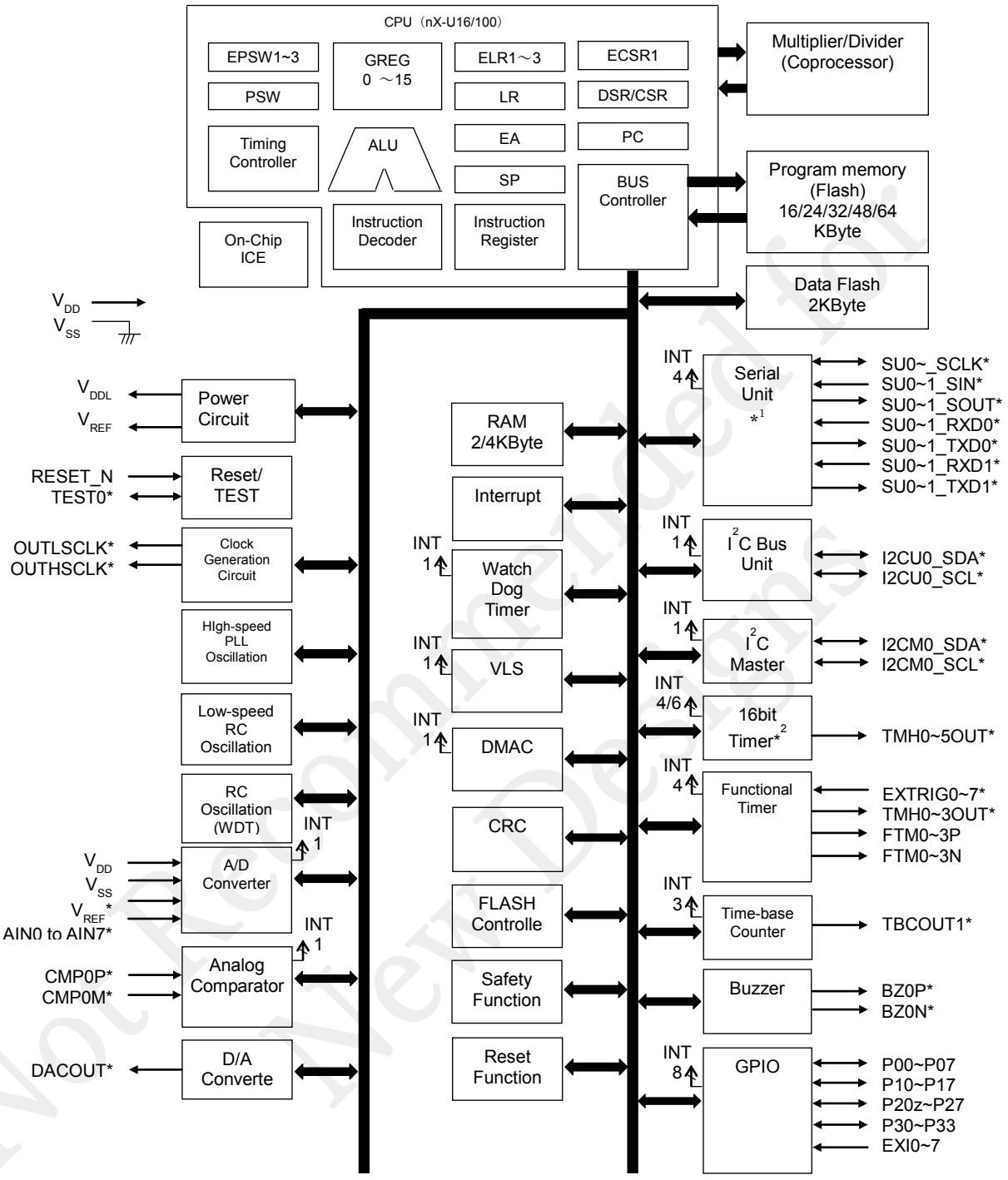
Table 1-2 Main Function List

Part number	Pin					Interrupt		Timer		Serial			Analog			
	Total pin-counts	Power pin counts	Input pin [RESET_N]	I/O port	LED drive port	Interna interrupt	External interrupt	Functional Timer [channel]	16bit General I Timer [channel] *1	Full-duplex UART or Synchronous serial [channel] *2	I²C bus unit (Master/Slave) [channel]	I²C bus interface (Master only) [channel]	10bit Sussesive type A/D converter [channel]	Analog comparator [channel]	Analog comparator [input pin]	8bit D/A converter [channel]
ML62Q1223A	16	3	1	12	11	22	8	4	4	2	1	1	6	1	2	0
ML62Q1224A																
ML62Q1225A																
ML62Q1233A	20			16	15								8			
ML62Q1234A																
ML62Q1235A																
ML62Q1245A	24			20	19	24			6							1
ML62Q1246A																
ML62Q1247A																
ML62Q1265A	32			28	27											
ML62Q1266A																
ML62Q1267A																

*1 : One 16bit timer is configurable as two 8bit timers
*2 : Full-duplex UART and Synchronous Serial Port can not be used simultaneously in the same channel. One Full-duplexUART is configurable as two half-duplex UARTs.

1.2 BLOCK DIAGRAM

1.2.1 Block Diagram of ML62Q1200A Group



* : indicates the 2nd to 8th functions.
*1 :Full-duplex x 2ch (One full-duplex UART is configurable as two half-dulplex UARTs).
*2 :Number of channel is dependent of part number. See Table 1 ML62Q1200A Group Main Function List.
*3 :Not available to use as a I/O port when connecting the on-chip debug emulator.

Figure 1-2 ML62Q1200A Group Block Diagram

1.3 PIN

1.3.1 Pin Layout

1.3.1.1 Pin Layout of ML62Q1223A/1224A/1225A 16pin SSOP Package

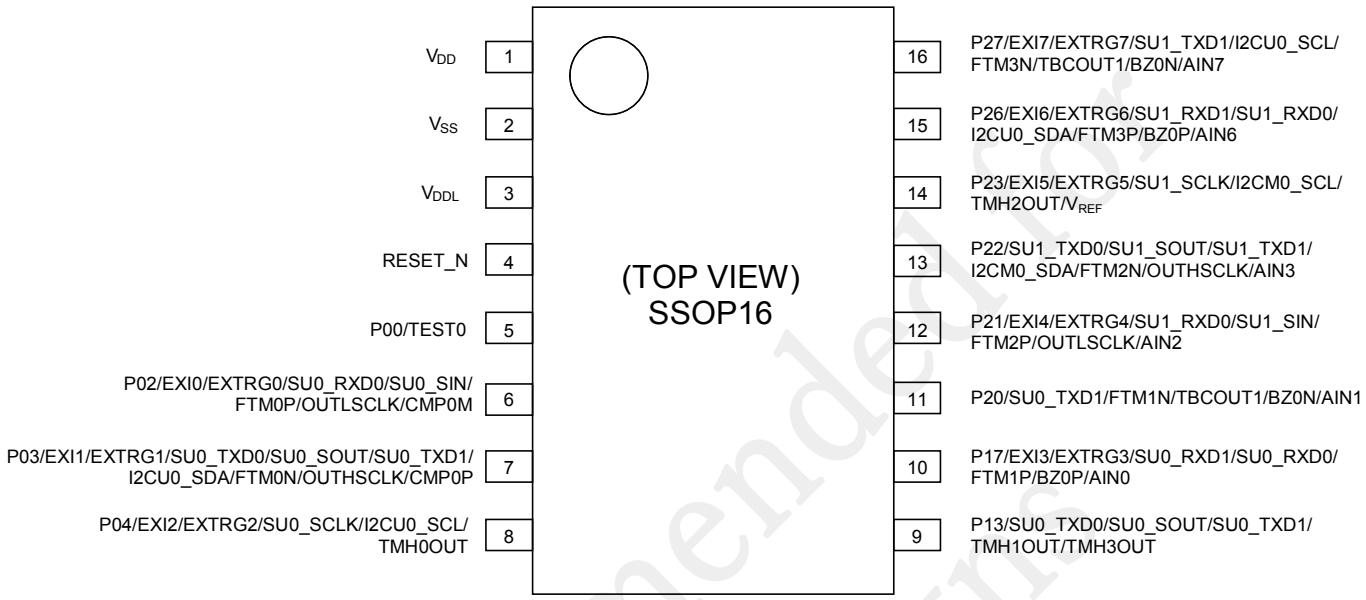


Figure 1-3 Pin Layout of ML62Q1223A/1224A/1225A 16pin SSOP Package

1.3.1.2 Pin Layout of ML62Q1223A/1224A/1225A 16pin WQFN Package

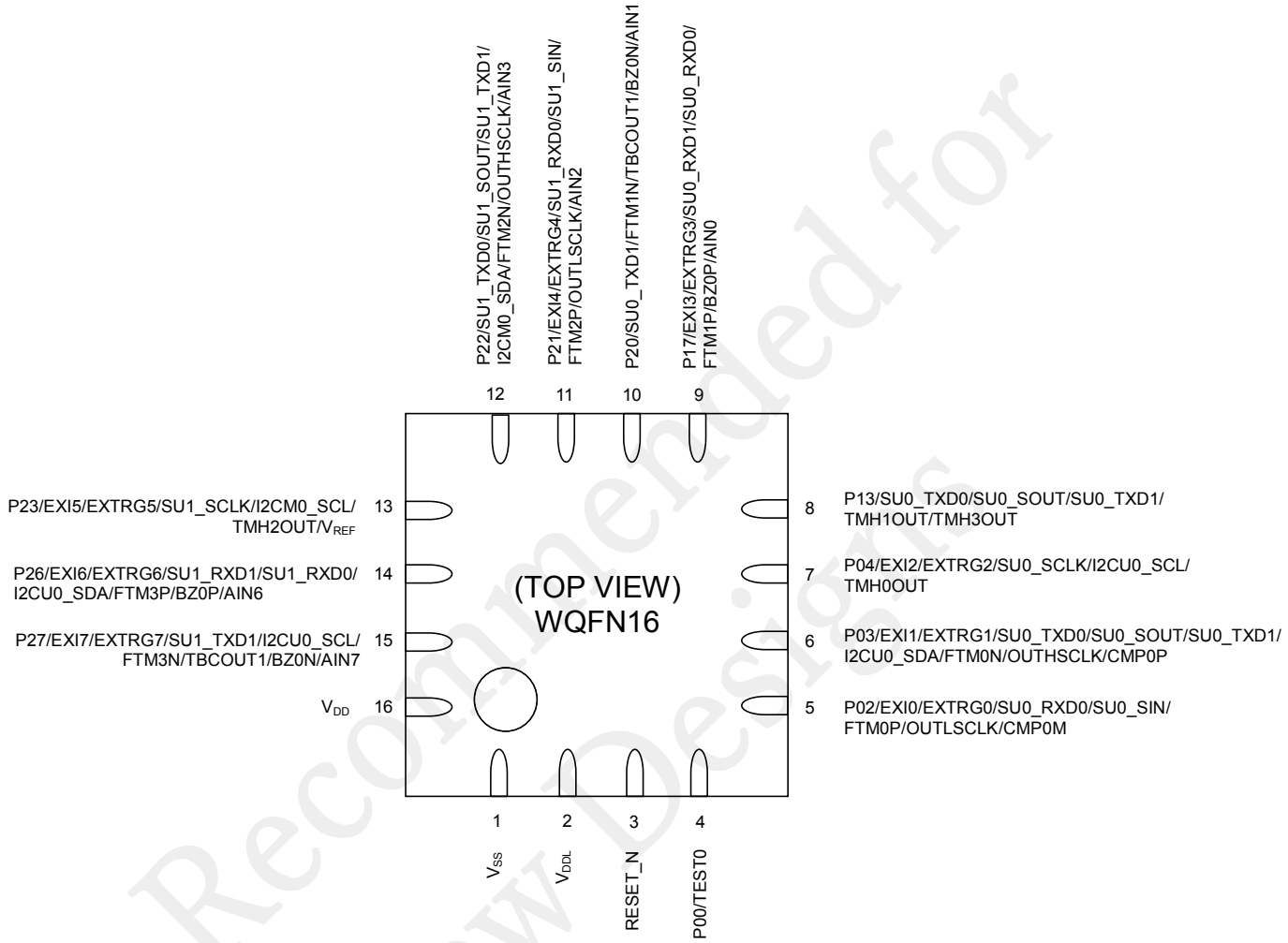


Figure 1-4 Pin Layout of ML62Q1223A/1224A/1225A 16pin WQFN Package

1.3.1.3 Pin Layout of ML62Q1233A/1234A/1235A 20pin TSSOP Package

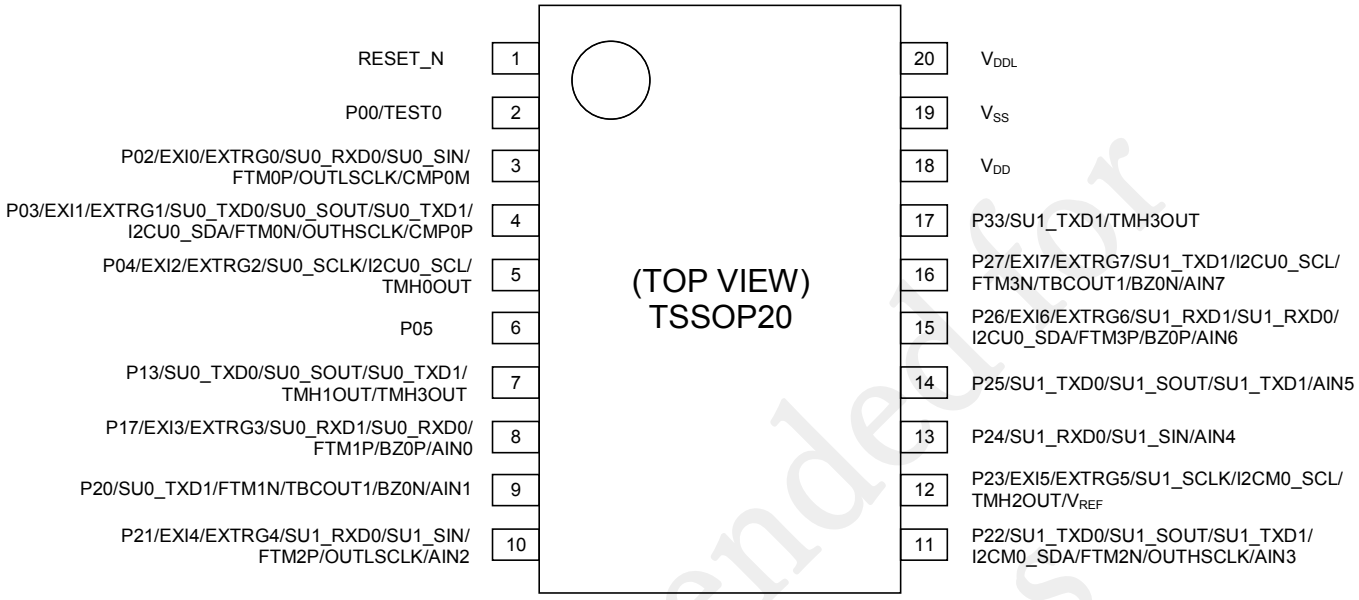


Figure 1-5 Pin Layout of ML62Q1233A/1234A/1235A 20pin TSSOP Package

1.3.1.4 Pin Layout of ML62Q1245A/1246A/1247A 24pin WQFN Package

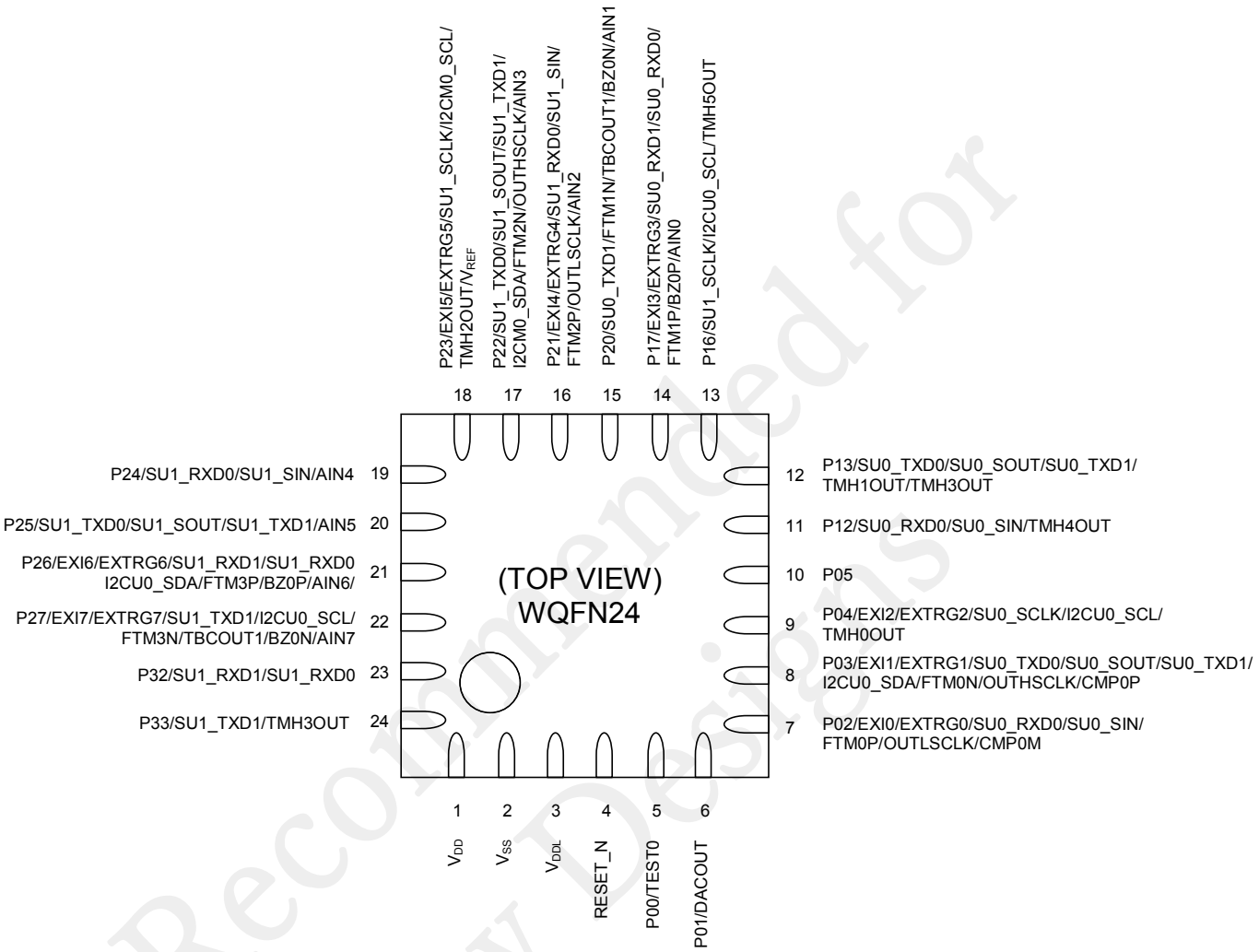


Figure 1-6 Pin Layout of ML62Q1245A/1246A/1247A 24pin WQFN Package

1.3.1.5 Pin Layout of ML62Q1265A/1266A/1267A 32pin TQFP Package

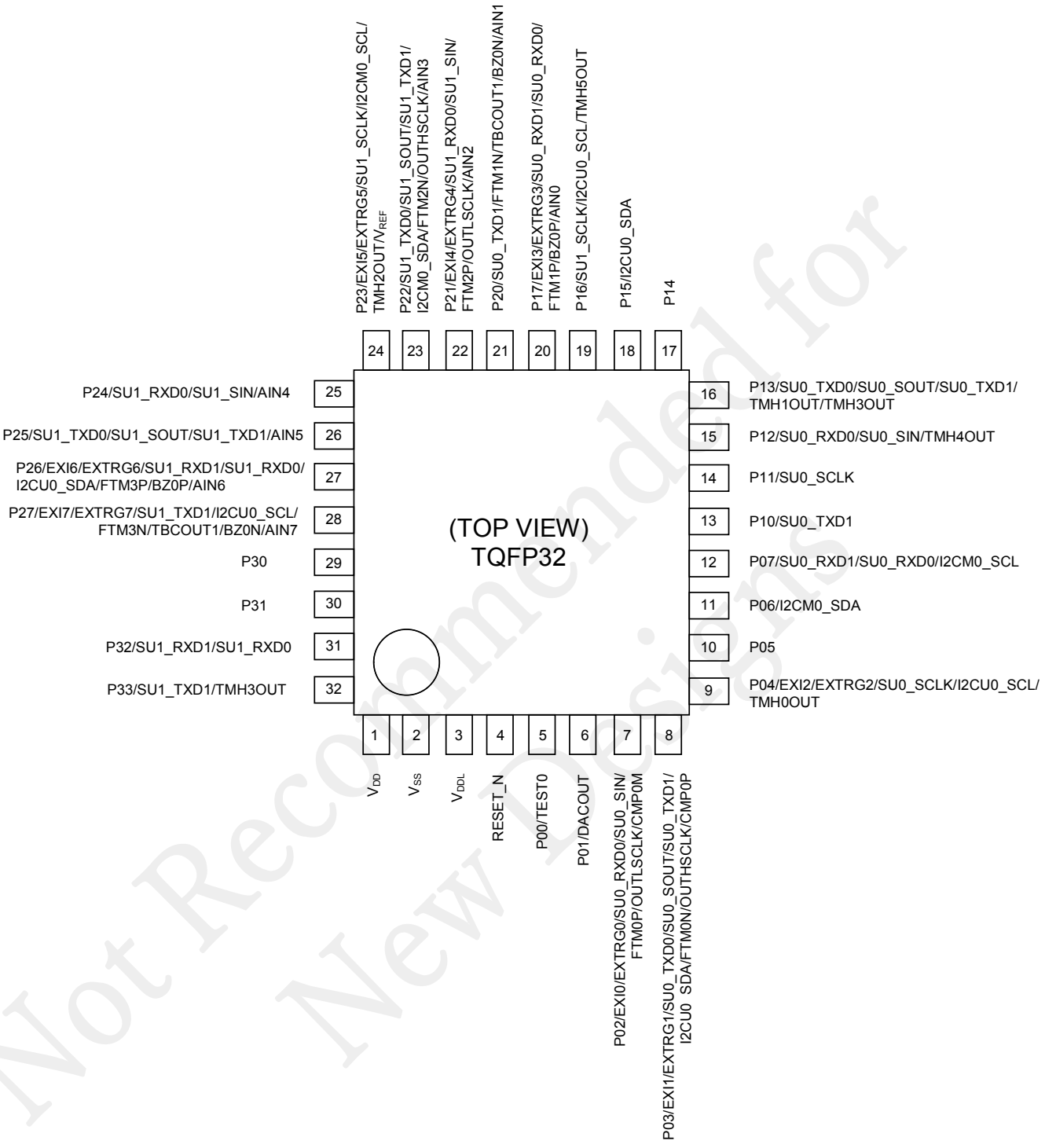


Figure 1-7 Pin Layout of ML62Q1265A/1266A/1267A 32pin TQFP Package

1.3.2 PIN LIST

Table 1-3 Pin List (1/4)

16Pin No. (SSOP)	16Pin No. (WQFN)	20Pin No.	24Pin No.	32Pin No.	Pn name	Primary function	Shared function	
1	16	18	1	1	V_{DD}	Positive power pin	—	
2	1	19	2	2	V_{SS}	Negative power pin	—	
3	2	20	3	3	V_{DDL}	Internal regulator output	—	
4	3	1	4	4	RESET_N (I)	Reset input Used for on-chip debug interface	—	
5	4	2	5	5	P00/TEST0 (I/O)	General I/O pin Used for on-chip debug interface (Not available to use as I/O pin when connecting to the on-chip emulator)	—	
-	-	-	6	6	P01 (I/O)	General I/O pin D/A converter output pin	2nd function	—
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—
6	5	3	7	7	P02/EXI0/EXTRG0 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU0_RXD0/SU0_SIN
							3rd function	—
							4th functiion	—
							5th function	FTM0P
							6th function	OUTLSCLK
							7th function	CMP0M
							8th function	—
7	6	4	8	8	P03/EXI1/EXTRG1 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU0_TXD0/SU0_SOUT
							3rd function	SU0_TXD1
							4th functiion	I2CU0_SDA
							5th function	FTM0N
							6th function	OUTHSCLK
							7th function	CMP0P
							8th function	—
8	7	5	9	9	P04/EXI2/EXTRG2 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU0_SCLK
							3rd function	—
							4th functiion	I2CU0_SCL
							5th function	TMH0OUT
							6th function	—
							7th function	—
							8th function	—
-	-	6	10	10	P05 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—

Table 1-3 Pin List (2/4)

16Pin No. (SSOP)	16Pin No. (WQFN)	20Pin No.	24Pin No.	32Pin No.	Pn name	Primary function	Shared function	
-	-	-	-	11	P06 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	I2CM0_SDA
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	-	12	P07 (I/O)	General I/O pin	2nd function	SU0_RXD1
							3rd function	SU0_RXD0
							4th functiion	I2CM0_SCL
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	-	13	P10 (I/O)	General I/O pin	2nd function	SU0_TXD1
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	-	14	P11 (I/O)	General I/O pin	2nd function	SU0_SCLK
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	11	15	P12 (I/O)	General I/O pin	2nd function	SU0_RXD0/SU0_SIN
							3rd function	—
							4th functiion	—
							5th function	TMH4OUT
							6th function	—
							7th function	—
							8th function	—
9	8	7	12	16	P13 (I/O)	General I/O pin	2nd function	SU0_TXD0/SU0_SOUT
							3rd function	SU0_TXD1
							4th functiion	—
							5th function	TMH1OUT
							6th function	—
							7th function	TMH3OUT
							8th function	—
-	-	-	-	17	P14 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—

Table 1-3 Pin List (3/5)

16Pin No. (SSOP)	16Pin No. (WQFN)	20Pin No.	24Pin No.	32Pin No.	Pn name	Primary function	Shared function	
-	-	-	-	18	P15 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	I2CU0_SDA
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	13	19	P16 (I/O)	General I/O pin	2nd function	SU1_SCLK
							3rd function	—
							4th functiion	I2CU0_SCL
							5th function	TMH5OUT
							6th function	—
							7th function	—
							8th function	—
10	9	8	14	20	P17/EXI3 /EXTRG3 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU0_RXD1
							3rd function	SU0_RXD0
							4th functiion	—
							5th function	FTM1P
							6th function	—
							7th function	BZ0P
							8th function	AIN0
11	10	9	15	21	P20 (I/O)	General I/O pin	2nd function	SU0_TXD1
							3rd function	—
							4th functiion	—
							5th function	FTM1N
							6th function	TBCOUT1
							7th function	BZ0N
							8th function	AIN1
12	11	10	16	22	P21/EXI4 /EXTRG4 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU1_RXD0/SU1_SIN
							3rd function	—
							4th functiion	—
							5th function	FTM2P
							6th function	OUTLSCLK
							7th function	—
							8th function	AIN2
13	12	11	17	23	P22 (I/O)	General I/O pin	2nd function	SU1_TXD0/SU1_SOUT
							3rd function	SU1_TXD1
							4th functiion	I2CM0_SDA
							5th function	FTM2N
							6th function	OUTHSCLK
							7th function	—
							8th function	AIN3
14	13	12	18	24	P23/EXI5 /EXTRG5 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU1_SCLK
							3rd function	—
							4th functiion	I2CM0_SCL
							5th function	TMH2OUT
							6th function	—
							7th function	—
							8th function	V_{REF}

Table 1-3 Pin List (4/5)

16Pin No. (SSOP)	16Pin No. (WQFN)	20Pin No.	24Pin No.	32Pin No.	Pn name	Primary function	Shared function	
-	-	13	19	25	P24 (I/O)	General I/O pin	2nd function	SU1_RXD0/SU1_SIN
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	AIN4
-	-	14	20	26	P25 (I/O)	General I/O pin	2nd function	SU1_TXD0/SU1_SOUT
							3rd function	SU1_TXD1
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	AIN5
15	14	15	21	27	P26/EXI6 / EXTRG6 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU1_RXD1
							3rd function	SU1_RXD0
							4th functiion	I2CU0_SDA
							5th function	FTM3P
							6th function	—
							7th function	BZ0P
							8th function	AIN6
16	15	16	22	28	P27/EXI7 / EXTRG7 (I/O)	General I/O pin External interrupt Functional timer external trigger	2nd function	SU1_TXD1
							3rd function	—
							4th functiion	I2CU0_SCL
							5th function	FTM3N
							6th function	TBCOUT1
							7th function	BZ0N
							8th function	AIN7
-	-	-	-	29	P30 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	-	-	30	P31 (I/O)	General I/O pin	2nd function	—
							3rd function	—
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—

Table 1-3 Pin List (5/5)

16Pin No.(SSOP)	16Pin No.(WQFN)	20Pin No.	24Pin No.	32Pin No.	Pn name	Primary function	Shared function	
-	-	-	23	31	P32 (I/O)	General I/O pin	2nd function	SU1_RXD1
							3rd function	SU1_RXD0
							4th functiion	—
							5th function	—
							6th function	—
							7th function	—
							8th function	—
-	-	17	24	32	P33 (I/O)	General I/O pin	2nd function	SU1_TXD1
							3rd function	—
							4th functiion	—
							5th function	TMH3OUT
							6th function	—
							7th function	—
							8th function	—

1.3.3 PIN DESCRIPTION

Table 1-4 Pin Description (1/3)

Function	Signal name	Pin name	I/O	Description	Logic
Power	–	V_{SS}	–	Negative power supply pin (-)	–
	–	V_{DD}	–	Positive power supply pin (+). Connect a capacitor C_V (1μF) between this pin and V_{SS}.	–
	–	V_{DDL}	–	Power supply pin for internal logic (internal regulator's output). Connect a capacitor C_V(1μF) between this pin and V_{SS}.	–
Test	TEST0	P00	I/O	Input pin for testing. Also, used for on-chip debug interface or ISP function. P00 is initialized as pull-up input mode by the system reset (not high-impedance mode).	Positive
System	V_{REF}	P23	–	Reference voltage output. An internal reference voltage in the SA type A/D converter block can be externally used for a reference. The pin is shared with the SA type A/D converter external reference voltage input.	–
	RESET_N	RESET_N	I	Input for reset. Asserting "L" level to this pin enters the MCU into system reset mode and internal circuits are initialized, then releasing it to "H" level make CPU start running the program. Used for on-chip debug interface or ISP function. Internal pull-up resistor is not installed.	Negative
	OUTLSCLK	P02 P21	O	Low-speed clock output.	–
	OUTHSCLK	P03 P22	O	Low-speed clock output.	–
General port (GPIO)	P00	P00	I/O	General I/O port - High-impedance - Input with Pull-UP (initial value) - Input without Pull-UP - CMOS output - N-channel open drain output P00 is only initialized as pulled-up input and other ports are initialized as high-impedence Not available to use as I/O pin when using for on-chip debug interface or ISP function.	Positive
	P01 - P07	P01 - P07	I/O	General I/O port - High-impedance (initial value) - Input with Pull-UP - Input without Pull-UP - CMOS output - N-channel open drain output	Positive
	P10 - P17	P10 - P17	I/O		Positive
	P20 - P27	P20 - P27	I/O		Positive
	P30 - P33	P30 - P33	I/O		Positive

Table 1-4 Pin Description (2/3)

Function	Signal name	Pin name	I/O	Description	Logic
UART	SU0_TXD0	P03	I/O	Serial communication unit0/UART0 data output pin.	Positive
		P13			
	SU0_RXD0	P02	I/O	Serial communication unit0/UART0 data input pin.	Positive
		P07			
		P12			
		P17			
	SU0_TXD1	P03	I/O	Serial communication unit0/UART1 data output pin.	Positive
		P10			
		P13			
		P20			
	SU0_RXD1	P07	I/O	Serial communication unit0/UART1 data input pin.	Positive
		P17			
	SU1_TXD0	P22	I/O	Serial communication unit1/UART0 data output pin	Positive
		P25			
	SU1_RXD0	P21	I/O	Serial communication unit1/UART0 data input pin.	Positive
		P24			
		P26			
		P32			
	SU1_TXD1	P22	I/O	Serial communication unit1/UART1 data output pin.	Positive
		P25			
		P27			
		P33			
	SU1_RXD1	P07	I/O	Serial communication unit1/UART1 data input pin.	Positive
		P26			
		P32			
Syncronous Serial Port	SU0_SIN	P02	I	Serial communication unit0/Synchronous serial data input pin.	Positive
		P12			
	SU0_SCK	P04	I/O	Serial communication unit0/Synchronous serial clock I/O pin.	Positive
		P11			
	SU0_SOUT	P03	O	Serial communication unit0/Synchronous serial data output pin.	Positive
		P13			
	SU1_SIN	P21	I	Serial communication unit1/Synchronous serial data input pin.	Positive
		P24			
	SU1_SCK	P16	I/O	Serial communication unit1/Synchronous serial clock I/O pin.	Positive
		P23			
	SU1_SOUT	P22	O	Serial communication unit1/Synchronous serial data output pin.	Positive
		P25			
I^2C Bus	I2CU0_SDA	P03	I/O	I^2C Unit0 (Master and Salve) Data I/O pin / N-ch open drain. Connect a pull-up resistor externally.	Positive
		P15			
		P26			
	I2CU0_SCL	P04	I/O	I^2C Unit0 (Master and Salve) Clock I/O pin / N-ch open drain. Connect a pull-up resistor externally.	Positive
		P16			
		P27			
	I2CM0_SDA	P06	I/O	I^2C Master0 Data I/O pin / N-ch open drain. Connect a pull-up resistor externally.	Positive
		P22			
	I2CM0_SCL	P07	I/O	I^2C Master0 Clock I/O pin / N-ch open drain. Connect a pull-up resistor externally.	Positive
		P23			

Table 1-4 Pin Description (3/3)

Function	Signal name	Pin name	I/O	Description	Logic
Functional Timer (FTM)	FTM0P	P02	O	Functional Timer0 output.	Positive
	FTM0N	P03	O	Functional Timer0 output.	Negative
	FTM1P	P17	O	Functional Timer1 output.	Positive
	FTM1N	P20	O	Functional Timer1 output.	Negative
	FTM2P	P21	O	Functional Timer2 output.	Positive
	FTM2N	P22	O	Functional Timer2 output.	Negative
	FTM3P	P26	O	Functional Timer3 output.	Positive
	FTM3N	P27	O	Functional Timer3 output.	Negative
	EXTRG0	P02	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG1	P03	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG2	P04	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG3	P17	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG4	P21	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG5	P23	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG6	P26	I	Functional Timer0-3 event trigger input pin.	—
	EXTRG7	P27	I	Functional Timer0-3 event trigger input pin.	—
16bit General Timer	TMH0OUT	P04	O	16bit General Timer 0 output pin	Positive
	TMH1OUT	P13	O	16bit General Timer 1 output pin	Positive
	TMH2OUT	P23	O	16bit General Timer 2 output pin	Positive
	TMH3OUT	P13 / P33	O	16bit General Timer 3 output pin	Positive
	TMH4OUT	P12	O	16bit General Timer 4 output pin	Positive
	TMH5OUT	P16	O	16bit General Timer 5 output pin	Positive
Time Base Counter (TBC)	TBCOUT1	P20 / P27	O	Time Base Counter 1Hz/2Hz output pin	Positive
Buzzer	BZ0P	P17 / P26	O	Buzzer output (positive phase)	Positive
	BZ0N	P20 / P27	O	Buzzer output (negative phase)	Negative
External Interrupt	EXI0	P02	I	GPIO maskable external interrupt pin	—
	EXI1	P03	I	GPIO maskable external interrupt pin	—
	EXI2	P04	I	GPIO maskable external interrupt pin	—
	EXI3	P17	I	GPIO maskable external interrupt pin	—
	EXI4	P21	I	GPIO maskable external interrupt pin	—
	EXI5	P23	I	GPIO maskable external interrupt pin	—
	EXI6	P26	I	GPIO maskable external interrupt pin	—
	EXI7	P27	I	GPIO maskable external interrupt pin	—
Successive approximation type A/D converter	V_{REFI}	P23	—	SA type A/D converter external reference voltage input. The voltage provided to the pin is used as the reference voltage for the A/D conversion.	—
	AIN0	P17	I	SA type A/D converter channel 0 input pin	—
	AIN1	P20	I	SA type A/D converter channel 1 input pin	—
	AIN2	P21	I	SA type A/D converter channel 2 input pin	—
	AIN3	P22	I	SA type A/D converter channel 3 input pin	—
	AIN4	P24	I	SA type A/D converter channel 4 input pin	—
	AIN5	P25	I	SA type A/D converter channel 5 input pin	—
	AIN6	P26	I	SA type A/D converter channel 6 input pin	—
	AIN7	P27	I	SA type A/D converter channel 7 input pin	—
Analog comparator	CMP0P	P03	I	Comparator input 0 (noninverting input)	—
	CMP0M	P02	I	Comparator input 0 (inverting input)	—
D/A converter	DACOUT	P01	O	D/A converter output pin	—

1.3.4 TERMINATION OF UNUSED PINS

Table 1-5 Termination of unused pins

Pin	Recommended pin termination
RESET_N	Connect to V_{DD} through a resistor
P00/TEST0	Open the pin with the internal initial condition of pulled-up input mode.
P01 to P07	Open the pins with the internal initial condition of Hi-impedance mode.
P10 to P17	
P20 to P27	
P30 to P33	

[Note]
For unused input ports or unused input/output ports, if an unstable middle level voltage is supplied to the corresponding pins which are configured as inputs without pull-up register or input/output mode, shoot-through may become excessively large. Therefore, it is recommended to configure those pins as either input mode with a pull-up resistor or output mode.

Chapter 2 CPU and Memory Space

2. CPU and Memory Space

2.1 General Description

ML62Q1000 series has LAPIS Semiconductor's original 16-bit CPU nX-U16/100(A35 core), a coprocessor for multiplication/division, flash memory in the program memory space, data flash and RAM in the data memory space and also has software remap function that remaps the program memory space by 4Kbyte unit.
Table 2-1 shows ML62Q1200A group's memory size in the program memory space and the data memory space.

Table 2-1 Memory Space of ML62Q1200A Group

Product name	Program memory space	Data memory space	
	Program memory size	Data flash size	RAM size
ML62Q1223A	16 Kbyte	2 Kbyte	2 Kbyte
ML62Q1224A	24 Kbyte	2 Kbyte	2 Kbyte
ML62Q1225A	32 Kbyte	2 Kbyte	2 Kbyte
ML62Q1233A	16 Kbyte	2 Kbyte	2 Kbyte
ML62Q1234A	24 Kbyte	2 Kbyte	2 Kbyte
ML62Q1235A	32 Kbyte	2 Kbyte	2 Kbyte
ML62Q1245A	32 Kbyte	2 Kbyte	4 Kbyte
ML62Q1246A	48 Kbyte	2 Kbyte	4 Kbyte
ML62Q1247A	64 Kbyte	2 Kbyte	4 Kbyte
ML62Q1265A	32 Kbyte	2 Kbyte	4 Kbyte
ML62Q1266A	48 Kbyte	2 Kbyte	4 Kbyte
ML62Q1267A	64 Kbyte	2 Kbyte	4 Kbyte

2.2 CPU nX-U16/100

The CPU nX-U16/100 in ML62Q1200A group is specified as SMALL model.
The nX-U16/100 has following features. For more details about nX-U16/100, see "nX-U16/100 Core Instruction Manual".

- Powerful Instruction Set
  - Instructions for data transfers, arithmetic, comparison, logic operations, 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 Spaces
  - Program/code memory (ROM)
  - Data memory (RAM)

- Interrupts
  - Dedicated emulator interrupts
  - Non-maskable interrupt
  - Maskable interrupts
  - Software interrupts

2.2.1 Wait mode and No wait mode

ML62Q1000 series has two CPU operating modes, wait mode and no wait mode. The mode is selectable by code option. See Chapter 26 "Code Option" for details on how to set the code option. The maximum CPU operating frequency in the wait mode and no wait mode is different. See the table 2-2.

Table 2-2 Maximum operating frequency of high-speed clock, peripheral and CPU

PLL oscillation mode	Maximum Operating Frequency		
	Peripheral	CPU (Wait mode)	CPU (No wait mode)
32MHz mode	32MHz	16MHz	8MHz
24MHz mode	24MHz	24MHz	12MHz
16MHz mode	16MHz	16MHz	8MHz

- Wait mode:

The instructions in Flash memory are stored into buffers and the CPU executes the instructions out of the buffers, which enables the instruction execution in high-speed operating frequency. The instructions are executed without a wait time in sequential instruction process, but it generates a wait time in branch instruction process(non sequential instruction process) and increases the execution cycles. The maximum operating frequency of CPU in the wait mode is 24MHz.

- No wait mode:

The CPU directry executes instructions out of the Flash memory. The instruction execution cycle is minimized. The maximum operating frequency of CPU in the no wait mode is 12MHz.

See Appendix C“ Instruction execution cycle” for details about the instruction execution cycle. The CPU mode is applicable even when the system clock is the low-speed clock (LSCLK).

2.3 Coprocessor

ML62Q1000 series has the built-in multiplication/division function in a coprocessor. The multiplier/divider is used by Coprocessor Data Transfer Instructions. For details about the instructions, see “nX-U16/100 Core Instruction Manual”.

2.3.1 Multiplication/Division

The multiplier/divider has following arithmetic functions.

- Multiplication: 16 bit x 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 calculation
- 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.

For the multiplication/division library that contains routine groups to execute operation by using this function, see "MULDIVU8LIB Multiplication and Division Library for U8/U16 Accelerator User’s Manual"

2.3.2 List of Registers

These are byte type registers for carrying out operations.

The registers use the byte length, however, they can be accessed as a word type register(ERn), double word type register (XRn), and quad word type register (QRn) when successive registers are combined.

Address	Function description of coprocessor general-purpose register	Symbol (Byte)	Symbol (Word)	Symbol (Double Word)	Symbol (Quad Word)	R/W	Initial value
-	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. The operation stats as soon as the data is written in CR7.

CR8 is a register to set the operation mode (signed, unsigned) and to enable operation. In the case of signed operation, the most significant bit is handled as a sign.

CR9 is a register to check the status of each operation result.

CR15 is the identification code register of the coprocessor. It has a fixed value.

[Note]

CR10 to CR14 have no function. Reading them gives "0x00." Writing is ignored.

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

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 symbol	CER0															
Byte symbol	CR1								CR0							
Bit symbol	areg15	areg14	areg13	areg12	areg11	areg10	areg9	areg8	areg7	areg6	areg5	areg4	areg3	areg2	areg1	areg0
Access type	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 symbol	CER2															
Byte symbol	CR3								CR2							
Bit symbol	breg15	breg14	breg13	breg12	breg11	breg10	breg9	breg8	breg7	breg6	breg5	breg4	breg3	breg2	breg1	breg0
Access type	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 symbol	CER4															
Byte symbol	CR5								CR4							
Bit symbol	creg15	creg14	creg13	creg12	creg11	creg10	creg9	creg8	creg7	creg6	creg5	creg4	creg3	creg2	creg1	creg0
Access type	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 symbol	CER6															
Byte symbol	CR7								CR6							
Bit symbol	dreg15	dreg14	dreg13	dreg12	dreg11	dreg10	dreg9	dreg8	dreg7	dreg6	dreg5	dreg4	dreg3	dreg2	dreg1	dreg0
Access type	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

A register to D register (CR0 to CR7) are registers to store the input values of operations and operation results. The bit symbol cannot be used in the software.

		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 × 16 bit＋32 bit	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]			
	Result	-		-		Multiply-accumulate [31:0]			
Multiply-accumulate (saturating) 16 bit × 16bit＋32 bit	Input	Multiplicand [15:0]		Multiplier [15:0]		Addend [31:0]			
	Result	-		-		Multiply-accumulate [31:0]			

In a saturating multiply-accumulate operation, the result is fixed to 0x7FFF_0xFFFF for a positive number and 0x8000_0x0000 for a negative number when it is out of the expressible range.

[Note]
"－" indicates that the previous value is retained.
In a signed operation, the most significant bits of input and output are signs.

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

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 symbol	CER8															
Byte symbol	CR9								CR8							
Bit symbol	c	z	s	ov	q	-	-	use	clen	-	-	sign	-	clmod2	clmod1	clmod0
Access type	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 operation mode register is a coprocessor general-purpose register to set the operation mode.

The operation status register is a coprocessor general-purpose register to store the states of the execution results of operations.

The bit symbol cannot be used in the software.

The values to be set to CR8 for execution of each operation mode are as follows.

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

Description of bits

- **clmod2 to clmod0** (Bits 2 to 0)

  clmod2, clmod 1, and clmod 0 are bits to select the operation mode. Multiplication, division, multiply-accumulate (non-saturating), and multiply-accumulate (saturating) are selectable.

CLMOD2	CLMOD1	CLMOD0	Description	
0	0	0	Multiplication (initial value)	16 bit * 16 bit
0	0	1	Division	32 bit/16bit
0	1	0	Multiply-accumulate (non-saturating)	16 bit * 16 bit + 32 bit
0	1	1	Multiply-accumulate (saturating)	16 bit * 16 bit + 32 bit
1	0	0	Reserved	No operation function
1	0	1	Division	32 bit / 32 bit
1	1	0	Reserved	No operation function
1	1	1	Reserved	No operation function

- **sign** (Bit 4)

  sign is a bit to set the sign operation.

SIGN	Description
0	Unsigned operation (initial value)
1	Signed operation

- **clen** (Bit 7)

  clen is a bit to set whether or not to enable the operation. When CLEN is set to "1", operation is enabled. If CLEN is cleared to be "0" during an operation, the next operation will not enabled after that operation is completed.

CLEN	Description
0	Operation disabled (initial value)
1	Operation enabled

- **use** (Bit 8)

  CLEN is a bit to set whether or not to enable the operation.
  When CLEN is set to "1", operation is enabled.
  If CLEN is cleared to be "0" during an operation, the next operation will not enabled after that operation is completed.

CLEN	Description
0	Operation disabled (initial value)
1	Operation enabled

- **q** (Bit 11)

  This becomes "1" for the saturated result of a saturating multiply-accumulate operation. The value is retained in the next operation. To set it to "0", it is necessary to write "0."

- **ov** (Bit 12)

  This becomes "1" if the result exceeds the range expressible by two's complement. This is set for each operation. Also, a value can be written.

- **s** (Bit 13)

  This becomes "1" when the 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 set for each operation. Also, a value can be written.

- **z** (Bit 14)

  This becomes "1" when the result is 0. The value is set for each operation. Also, a value can be written.

- **c** (Bit 15)

  This becomes "1" when the result is carried or the divisor is 0 in the division mode. The value is set for each operation. Also, a value can be written.

The flags changes during each operation as follows:

	sign	c	z	s	ov	q
Multiplication 16 bit x 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 x 16 bit + 32 bit	1 (signed)	*	*	*	*	-
	0 (unsigned)	*	*	*	*	-
Multiply-accumulate (saturating) 16 bit x 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)

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 symbol	CER14															
Byte symbol	CR15								CR14							
Bit symbol	COPID7	COPID6	COPID5	COPID4	COPID3	COPID2	COPID1	COPID0	-	-	-	-	-	-	-	-
Access type	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

CR15 is a read-only coprocessor general-purpose register to indicate the coprocessor ID. The data of CR15 is fixed to "0x81".

The bit symbol cannot be used in the software.

2.3.3 Description of Operation

When using the multiplication/division function, see "MULDIVU8LIB Multiplication and Division Library for U8/U16 Accelerator User's Manual" for the available multiplication/division library.

Figure 2-1 shows an example of software to execute multiplication without using the library.

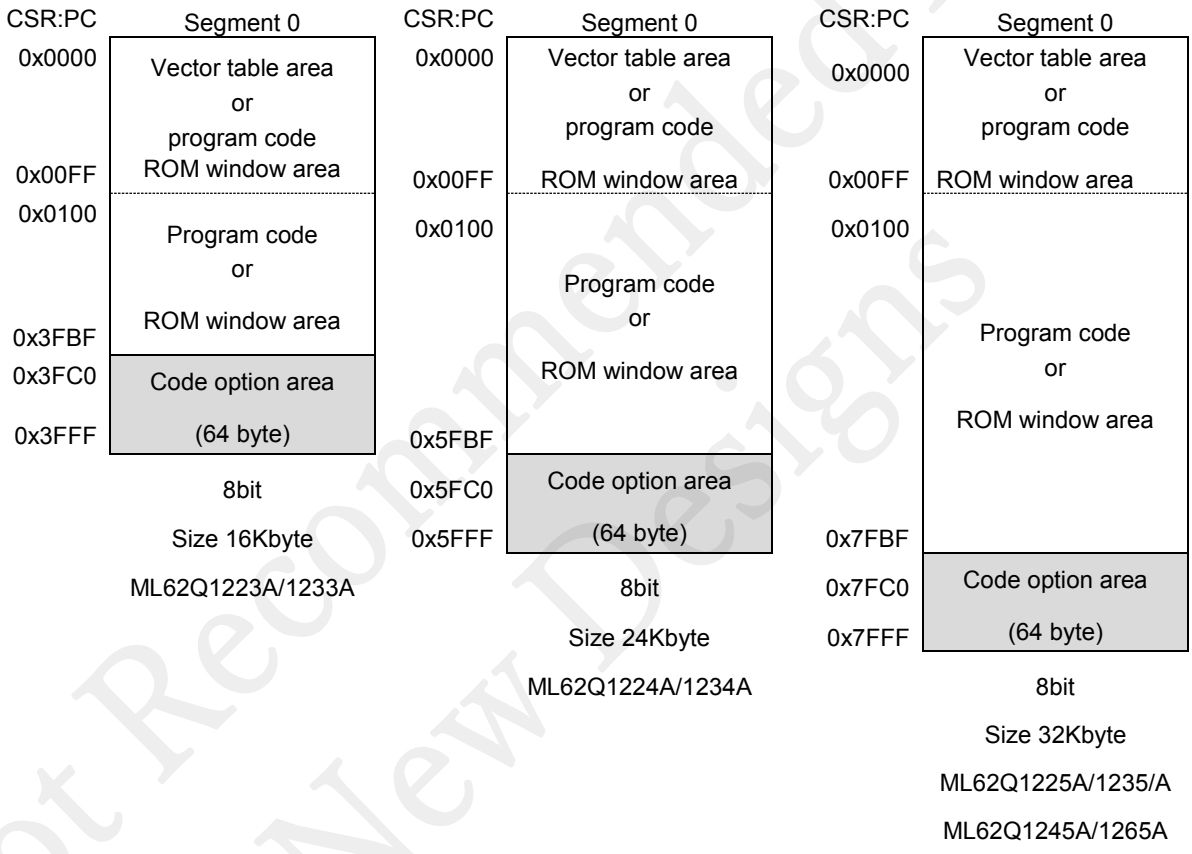
```
; 0x1234 × 0x0AA55 multiplication example

    #asm
            mov     r0  ,   #90h    ; Set the multiplier mode
            mov     cr8,    r0      ; Set signed multiplier mode
            ;
            mov     r2,     #55h    ; Set the multiplier
            mov     r3,     #0aah   ;
            mov     r0,     #34h    ; Set the multiplicand
            mov     r1,     #12h    ;
            ;
            mov     cr4,    r0      ; Transfer multiplier [7:0]
            mov     cr5,    r1      ; Transfer the multiplier [15:8]
            mov     cr6,    r2      ; Transfer multiplicand [7:0]
            mov     cr7,    r3      ; Transfer multiplicand [15:8], start calculation
            ;
            nop                     ; Wait for calculation (4 clock)
            nop                     ;
            nop                     ;
            nop                     ; Calculation end
            ;
            mov     r0,     cr0     ; Transfer product [7:0]
            mov     r1,     cr1     ; Transfer product [15:8]
            mov     r2,     cr2     ; Transfer product [23:16]
            mov     r3,     cr3     ; Transfer product [31:24]
        #endasm
```

Figure 2-1 Software Example for Multiplication without Using Library

2.4 Program Memory Space

The program memory space is an area to store the program code, table data (ROM window) or vector table.
The program stores 16-bit data and is specified by 20 bits (CSR: PC) consisting of higher 4 bits as code segment register (CSR) and lower 16 bits as PC (program counter).
The ROM window area is 8-bit data that can be used as table data.
The vector table, which has 16-bit long data, can be used as reset vectors, hardware interrupt vectors, and software interrupt vectors. Unused vector area is available to use as a program code area. The code option area is used to select CPU operating mode, PLL oscillation frequency and etc.
Figure 2-2 shows the program memory space configuration.



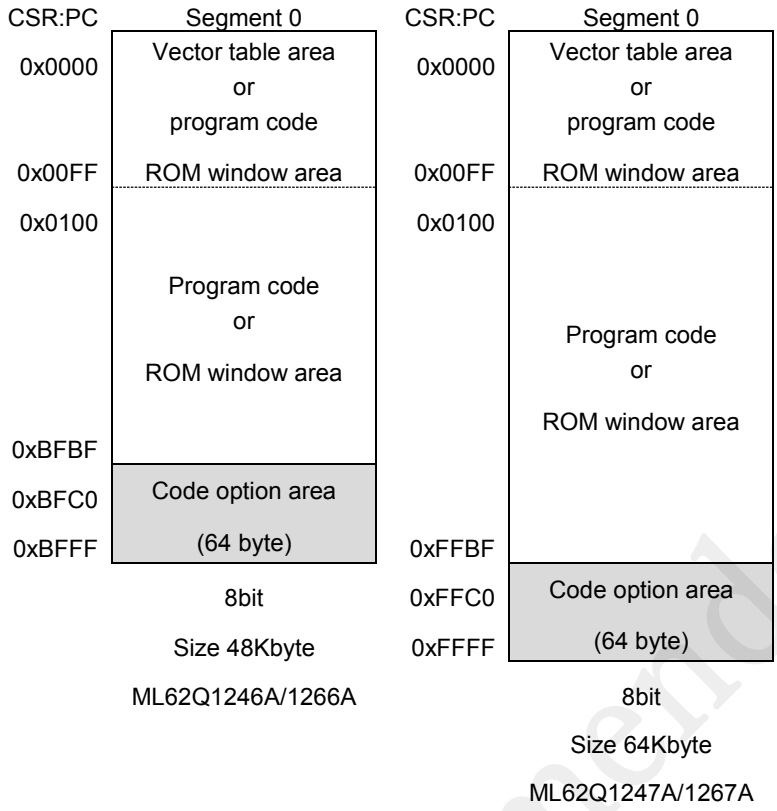


Figure 2-2 Program Memory Space Configuration

[Note]

- The code option area(64 bytes) is not available for program code area. For details of the code option area, see Chapter 26 "Code Option" and make sure setting data.
- It is recommended for failsafe to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space by using HTU8. See “HTU8 User’s Manual” and “nX-U16/100 Core Instruction Manual”
- Do not access unused area to avoid the CPU runs in correct.

2.5 Data Memory Space

The data memory space of this LSI consists of the ROM window area, RAM area, and SFR area of Segment 0 and the ROM reference area of Segment 8.

The data memory stores 8-bit data and is specified by 21 bits consisting of higher 5 bits as DSR and lower 16 bits as addressing specified by each instruction.

Figure 2-3 ~ Figure 2~8 show the data memory space configuration of ML62Q1200A group. Other segments not shown in the fugure are unused area.

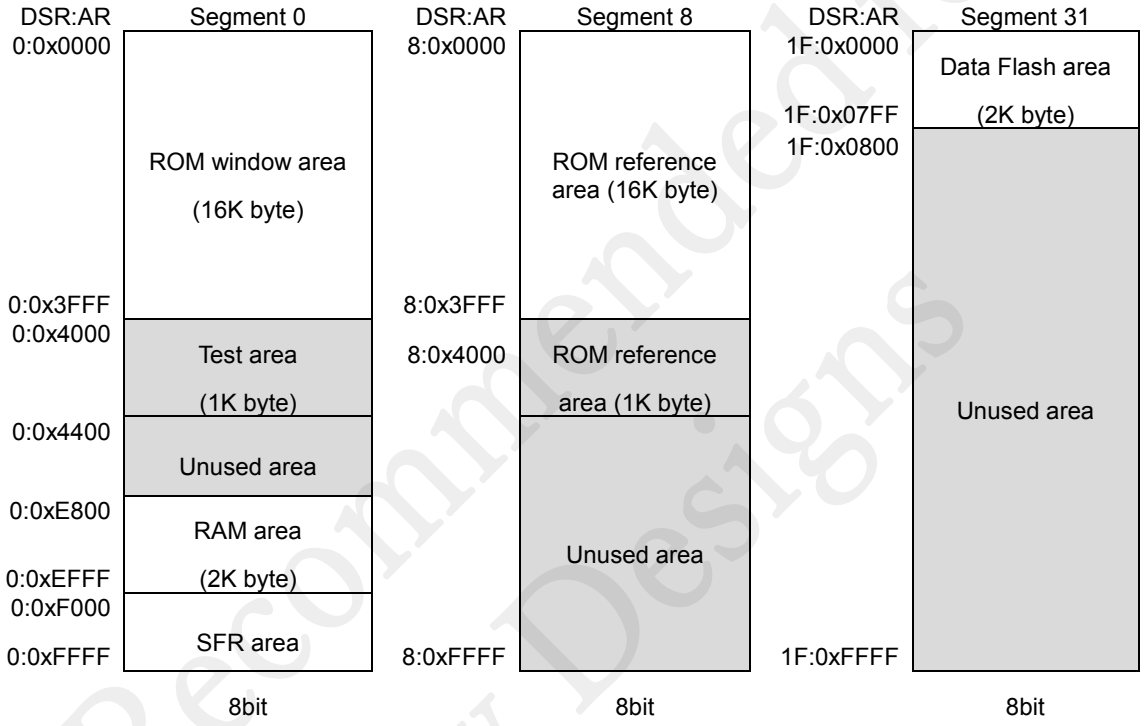


Figure 2-3 ML62Q1223A/ML62Q1233A Data Memory Space Configuration

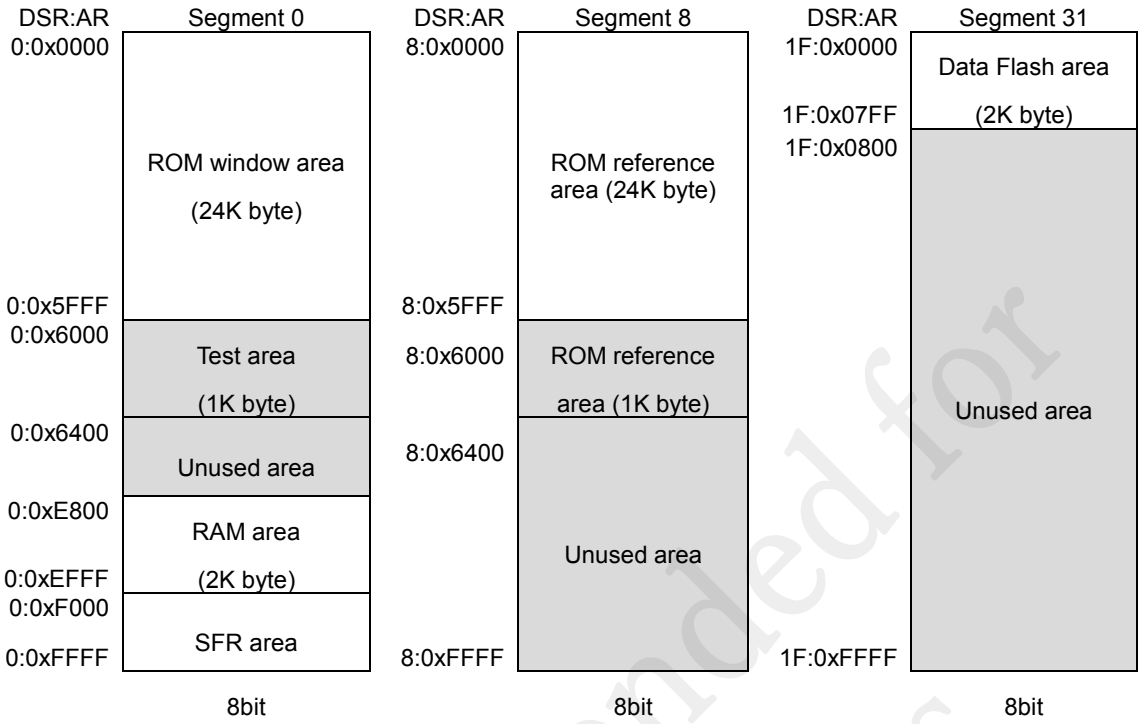


Figure 2-4 ML62Q1224A/ML62Q1234A Data Memory Space Configuration

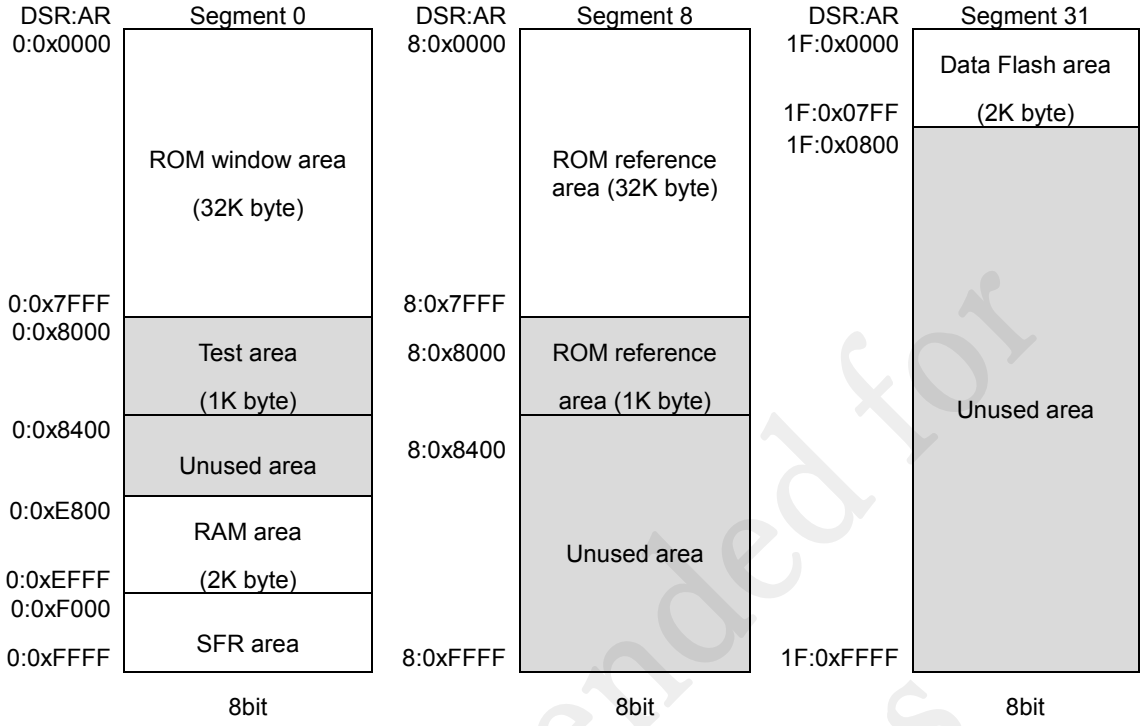


Figure 2-5 ML62Q1225A/ML62Q1235A Data Memory Space Configuration

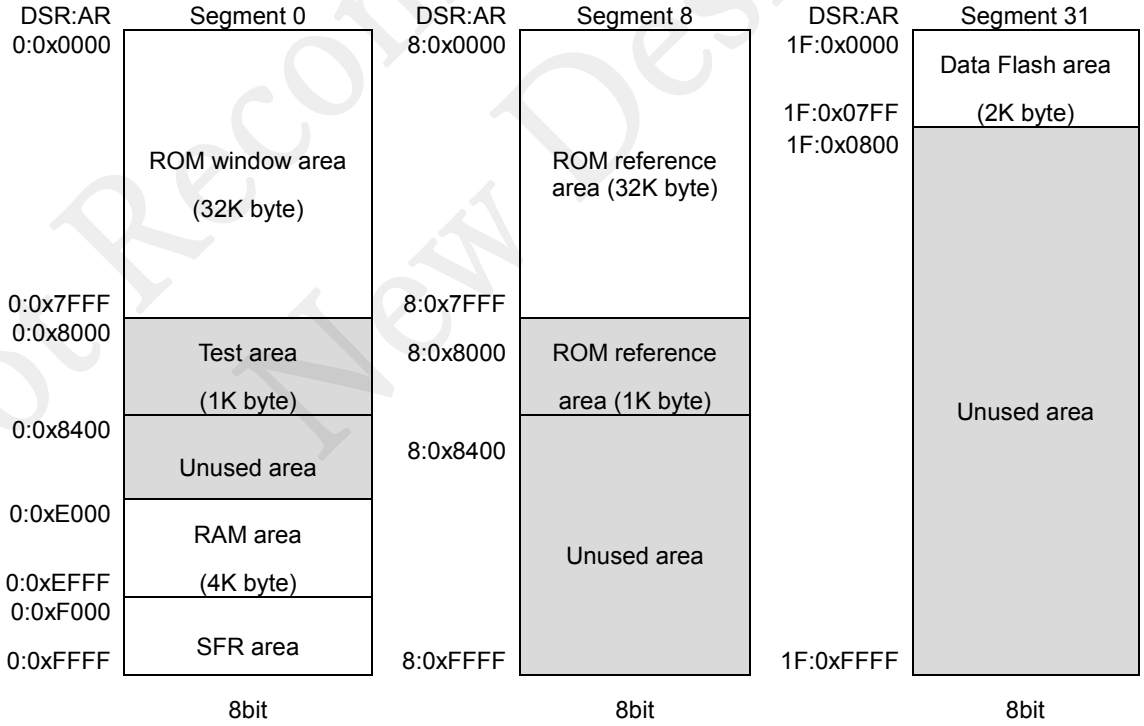


Figure 2-6 ML62Q1245A/ML62Q1265A Data Memory Space Configuration

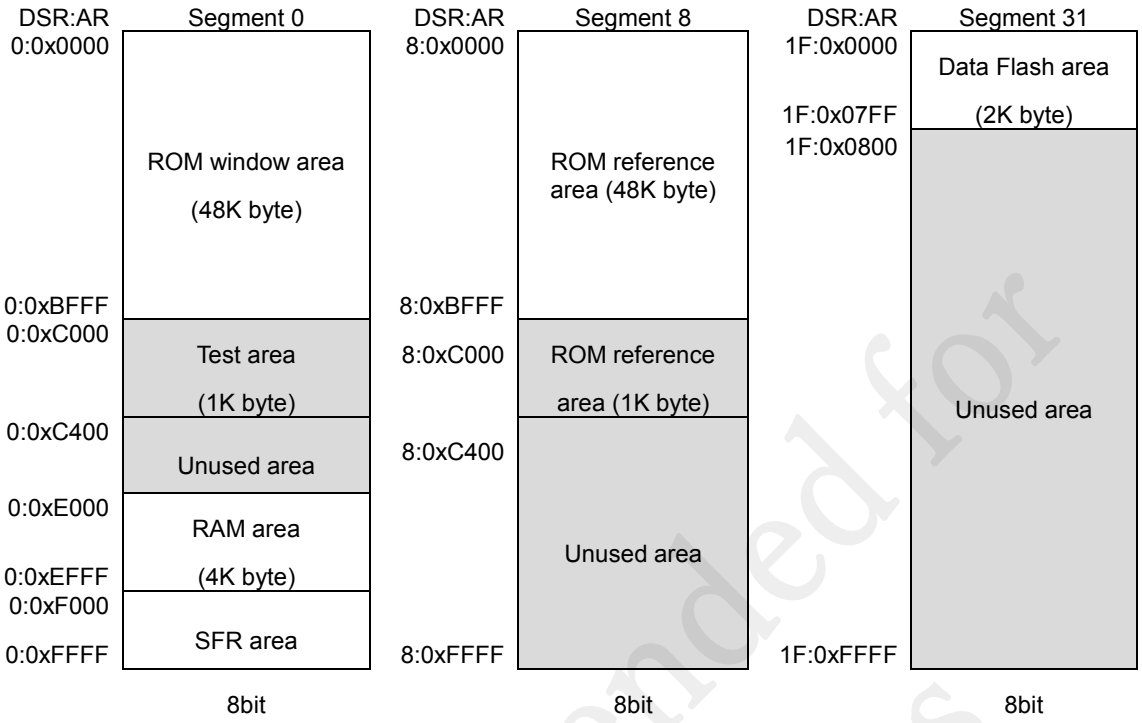


Figure 2-7 ML62Q1246A/ML62Q1266A Data Memory Space Configuration

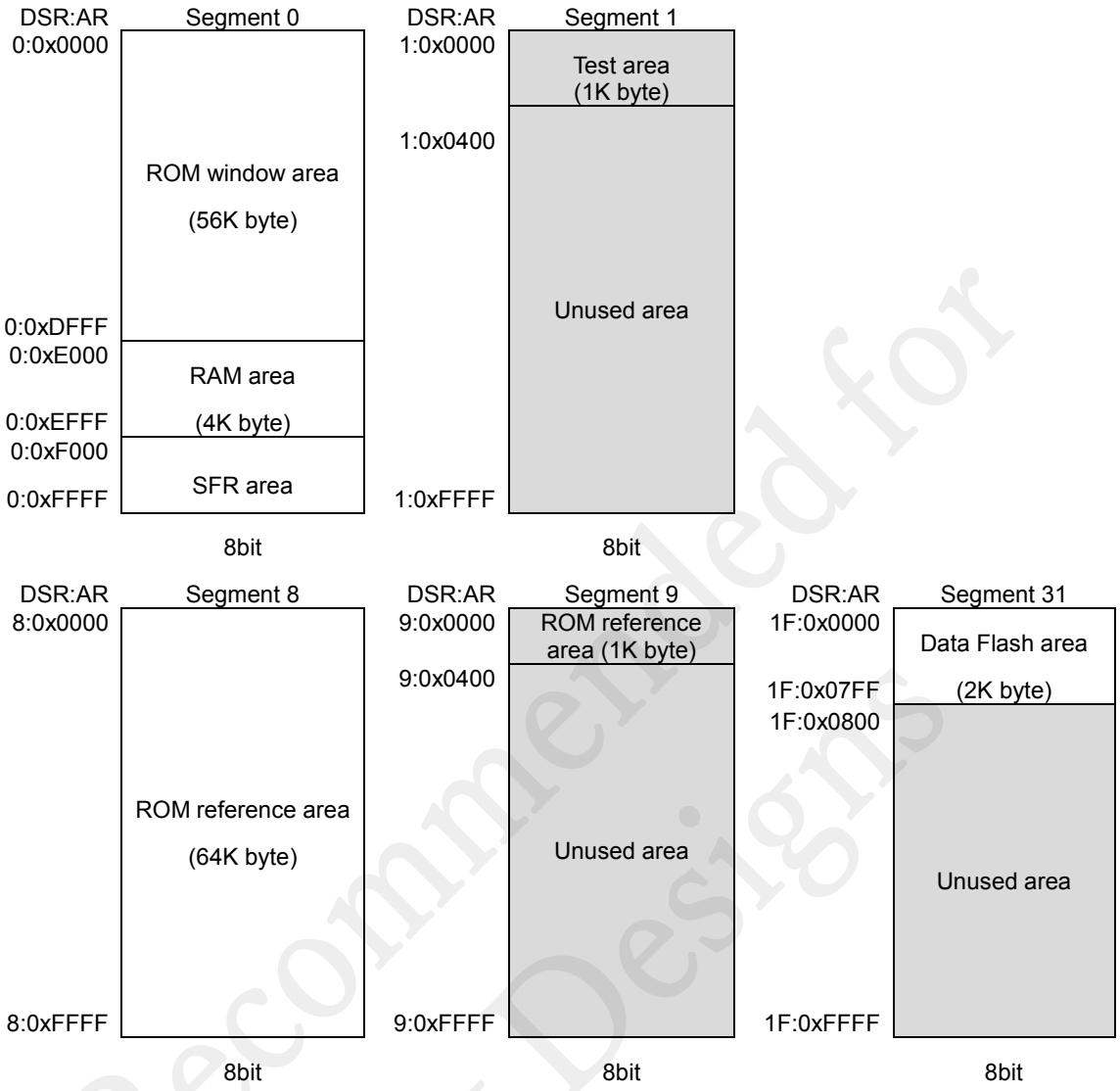


Figure 2-8 ML62Q1247A/ML62Q1267A Data Memory Space Configuration

[Note]

- The contents of RAM area are undefined at power-up and system reset. Initialize the area by thesoftware.
- The segment 0 of program memory space and the segment of data memory space is in different, but the segment 0 of program memory space is readable through the ROM window area of the data memory space.
- The segment 8 and 9 is mirror area of segment 0 and 1 in the program memory space. The contents of Segment 0 of the program memory space is readable from the ROM reference area of Segment 8. The contents of Segment 1 of the program memory space is readable from the ROM reference area of Segment 9.
- The test area 1K byte(512 word) has MCU's unique data and the area cannot be used as the program memory.
- Do not access unused area to prevent the CPU running incorrctly.

2.6 Description of Registers

2.6.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF000	Data segment register	DSR	-	R/W	8	0x00
0xF0A0	Flash remap address register	REMAPADD	-	R/W	8	0x00

2.6.2 Data Segment Register (DSR)

Address: 0xF000

Access: R/W

Access size: 8 bits

Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								DSR							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	DSR4	DSR3	DSR2	DSR1	DSR0
Access type	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

DSR is a special function register (SFR) used to retain a data segment. For details of DSR, see "nX-U16/100 Core Instruction Manual".

Description of bits

- **DSR4 to DSR0** (Bit 4 to 0)

DSR4	DSR3	DSR2	DSR1	DSR0	Description
0	0	0	0	0	Data segment 0 (initial value)
0	0	0	0	1	Data segment 1 (disabled)
0	0	0	1	0	Data segment 2 (disabled)
0	0	0	1	1	Data segment 3 (disabled)
0	0	1	0	0	Data segment 4 (disabled)
0	0	1	0	1	Data segment 5 (disabled)
0	0	1	1	0	Data segment 6 (disabled)
0	0	1	1	0	Data segment 7 (disabled)
0	1	0	0	0	Data segment 8 (mirror of Segment 0)
0	1	0	0	1	Data segment 9 (mirror of Segment 1)
0	1	0	1	0	Data segment 10 (disabled)
0	1	0	1	1	Data segment 11 (disabled)
0	1	1	0	0	Data segment 12 (disabled)
0	1	1	0	1	Data segment 13 (disabled)
0	1	1	1	0	Data segment 14 (disabled)
0	1	1	1	1	Data segment 15 (disabled)
▪	▪	▪	▪	▪	Data segments 16 to 30 (disabled)
1	1	1	1	1	Data segment 31 (data FLASH)

[Note]
0xFF is read from unused area of data segment.

2.6.3 Flash Remap Register (REMAPADD)

Address: 0xF0A0

Access: R/W

Access size: 8 bits

Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								REMAPADD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	REA15	REA14	REA13	REA12
Access type	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

REMAPADD is a special function register (SFR) used to specify the remap area.
The addresses 0000H to 0FFFH (4 Kbyte) are remapped from the addresses specified to REA15 to REA12 bit to the area of the same size within 4 Kbyte. The remap area can be set only for Segment 0.

When the software reset is executed by the BRK instruction (* only the CPU is reset), the CPU executes instructions from the beginning address of the remap area specified in REMAPADD. With the remap function, all the vector table areas (reset vector, hardware interrupt vector, and software interrupt vector area) are remapped. * For the BRK instruction, see "nX-U16/100 Core Instruction Manual."

Description of bits

- **REA15 to REA12** (Bit 3 to 0)
  REA15 to REA12 are bits to set the higher 4 bits (bit 15 to 12) of the beginning address of the area to be remapped.
  Example) When "0FH" is written in REA15 to 12 and the BRK instruction is executed, the area of 0x0F000 to 0x0FFFF is mapped to 0x0000 to 0x0FFF.

2.7 Software Remap Function

The self-programming function for updating the software (IAP: In-Application Programming) is avaiable using the software remap. The software remap function remaps the “4Kbyte area which the start address is set in REMAPADD register” to the area of addresses 0x0000 to 0x0FFF (4 Kbyte).

The remap function also remaps the vector table areas (reset vector, hardware interrupt vector, and software interrupt vector areas). The software remap function can be enabled/disabled by the code option. For details of code options, see Chapter 26 "Code Option."

Set a a start address of remap target area to REMAPADD register, write “2” to ELEVEL of CPU program status word(PSW) and execute BRK instruction. Then, executing the BRK instruction resets only CPU and the CPU exceutes instructions from the start address set in the REMAPADD register. For details about the BRK instruction and ELEVEL, see "nX-U16/100 Core Instruction Manual".

Figure 2-9 shows an example of software for remapping (when the beginning address for remapping is 0x0F000).

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

Figure 2-9 Example of Software for Remapping

Figure 2-10 shows the memory map before and after remapping.
After remapping, read from Segment 8 to read the area of 0:0x0000 to 0:0x0FFF (4 Kbyte) before remapping.

In the case to rewrite the area of 0:0x0000 to 0:0x0FFF (4 Kbyte) after remapping, set the addresses (0:0000H to 0:0FFFH) to flash address register (FLASHA). If need to read the noarml boot area(before remapping) of 0:0x0000, read from Segment 8 to read the area(mirror area of segment 0). Codes in the remapped area are read from the segment 0. See Chapter 25 "Flash Memory" for details about the flash address register (FLASHA)." See another document "ML62Q1000 series IAP sample program" for more details on how to use the software remap function.

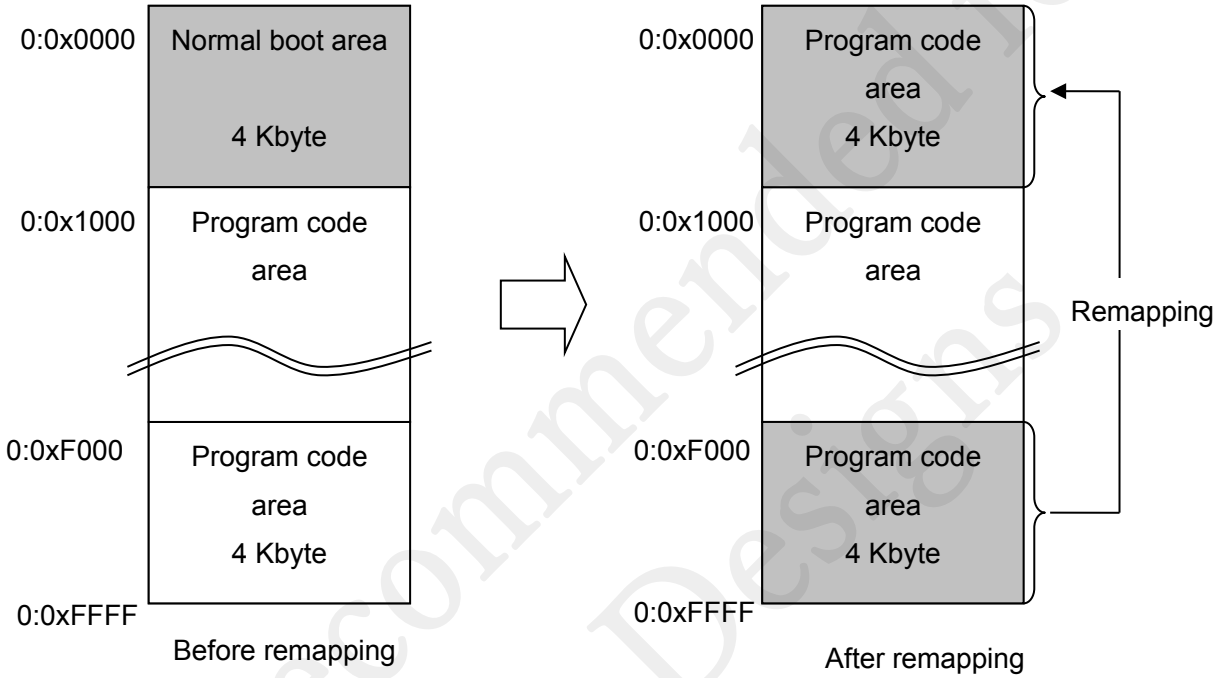


Figure 2-10 Memory Map Before and After Remapping of Program Memory Space (64 Kbyte)

Chapter 3 Reset Function

3. Reset Function

3.1 General Description

ML62Q1000 series has reset generation functions.
Following causes make the resets to the CPU and peripherals and SFRs. This chapter describes about the system reset modes, RESET_N pin reset and Power-on reset. For other resets, see each of the related chapter.
See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

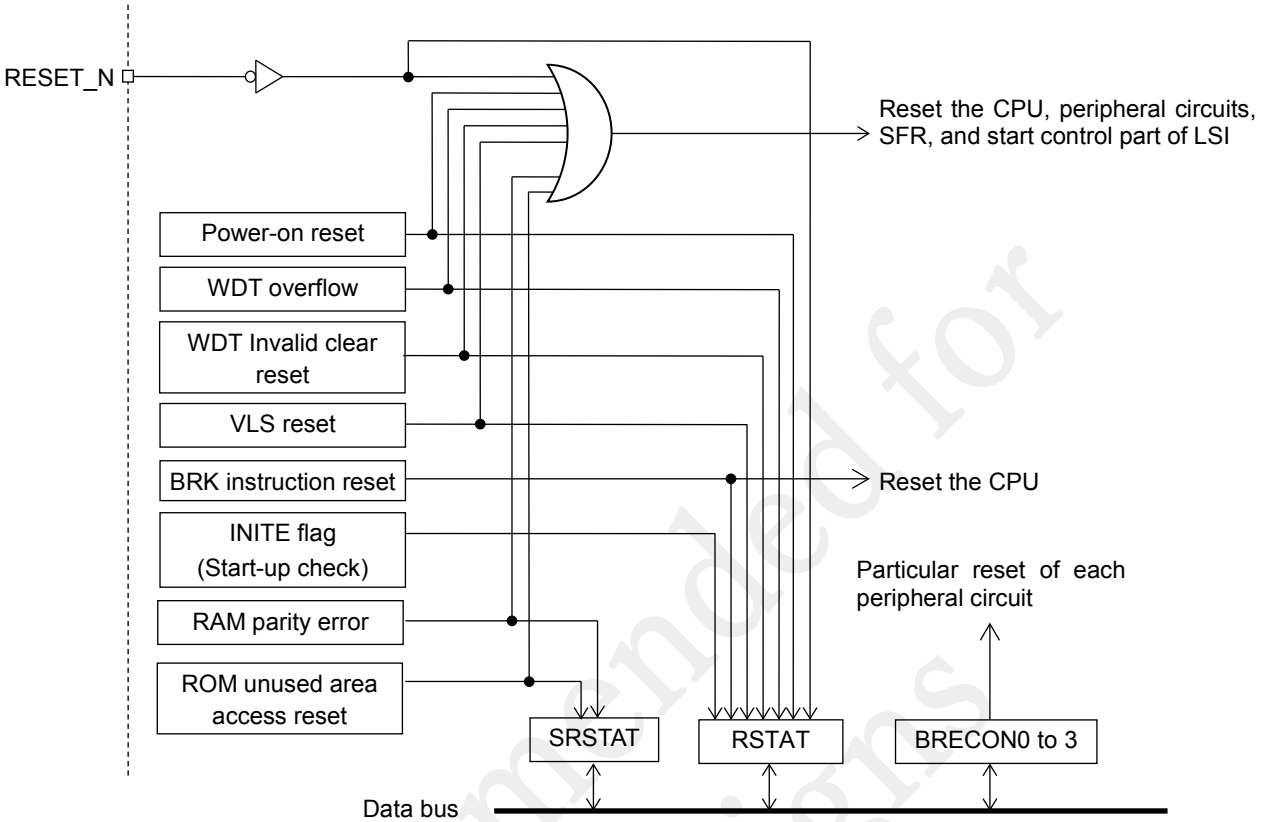
- Reset by the RESET_N pin
- Reset by power-on detection
- Reset by VLS (Voltage Level Supervisor)
- Reset by the 2nd WDT (watchdog timer) overflow
- Reset by WDT counter invalid clear operation
- Reset by RAM parity error
- Reset by ROM unused area access
- CPU reset by executing the BRK instruction (when ELEVEL of PSW = 2 or larger)
- Reset of peripheral circuits by block reset control registers (BRECON0 to 3)

3.1.1 Features

- Reset status register (RSTAT) indicates the reset generation causes
- Safety function reset status register (SRSTAT) indicates the reset generation causes
- INITE flag indicates 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
BRECON0 to 3 : Block reset control register 0 to 3

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 (Byte)	Symbol (Word)	R/W	Size	Initial value
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

[Note]
For registers with word symbol, word access is possible. For word access, specify an even address.

3.2.2 Reset Status Register (RSTAT)

Address: 0xF058
Access: R/W
Access size: 8 bits/16bits
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	RSTAT															
Byte symbol	RSTATH								RSTATL							
Bit symbol	-	-	-	-	-	-	-	BRKR	INITE	RSTR	-	VLS0R	WDTWR	WDTR	-	POR
Access type	-	-	-	-	-	-	-	R/W	R	R/W	R/W	R/W	R/W	R/W	-	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

RSTAT is a specific function register (SFR) to indicate the cause of occurrence of a reset.
When a reset occurs, only the bit that indicates the cause of the reset occurred is set to "1". Other bits (except the INITE bit) retain values before occurrence of the reset. After identifying the cause of the reset, write "0xFFFF" to the RSTAT register to initialize to “0x0000”it for preparing the next reset.

- Description of bits**POR** (Bit 0)
  POR is a flag to indicate that a power-on reset has occurred.
  The flag is cleared when "1" is written.

POR	Description
0	Power-on reset not occurred
1	Power-on reset occurred (initial value)

- **WDTR** (Bit 2)
  WDTR is a flag to indicate that a reset by the second overflow of the watchdog timer has occurred.
  The flag is cleared when "1" is written.

WDTR	Description
0	Watchdog timer reset not occurred (initial value)
1	Watchdog timer reset occurred

- **WDTWR** (Bit 3)
  WDTWR is a flag to indicate that a reset by watchdog counter clear operation occurred while the window of the watchdog timer was closed. The flag is cleared when "1" is written.

WDTWR	Description
0	Reset not occurred with clear operation while the window of the watchdog timer was closed (initial value)
1	Reset occurred with clear operation while the window of the watchdog timer was closed

- **VLS0R** (Bit 4)
  VLS0R is a flag to indicate that voltage level detection reset has occurred.
  The flag is cleared when "1" is written.

VLS0R	Description
0	Voltage level detection reset (VLS0) not occurred (initial value)
1	Voltage level detection reset (VLS0) occurred

- **RSTR** (Bit 6)
  RSTR is a flag to indicate that a RESET_N pin reset has occurred.
  The flag is cleared when "1" is written.

RSTR	Description
0	RESET_N pin reset not occurred
1	RESET_N pin reset occurred

- **INITE** (Bit 7)
  INITE is a flag to indicate that the LSI has been started normally or not.
  In the case the INITE is set to “1”, it will be reset by the RESETN pin reset or Power-On-Reset.

INITE	Description
0	LSI started-up normally (initial value)
1	Any error occurred during the start-up

- **BRKR** (Bit 8)
  BRKR is a flag to indicate that a CPU reset by BRK instruction has occurred.
  The flag is cleared when "1" is written.

BRKR	Description
0	CPU reset by BRK instruction not occurred (initial value)
1	CPU reset by BRK instruction occurred

3.2.3 Safety Function Reset Status Register (SRSTAT)

Address: 0xF05A
Access: R/W
Access size: 8 bits
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								SRSTAT							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RPER	FIAR
Access type	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

SRSTAT is a special function register (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.

Description of bits

- **FIAR** (Bit 0)
  FIAR is a flag to indicate that ROM unused area access reset occurred.
  The flag is cleared when "1" is written.

FIAR	Description
0	ROM unused area access reset not occurred (initial value)
1	ROM unused area access reset occurred

- **RPER** (Bit 1)
  RPER is a flag to indicate that a RAM parity error reset has occurred.
  The flag is cleared when "1" is written.

RPER	Description
0	RAM parity error reset not occurred (initial value)
1	RAM parity error reset occurred

3.3 Description of Operation

3.3.1 Operation in Reset Mode

The ML62Q1200A Group can execute reset operation with the following causes.

- Reset by the RESET_N pin
- Reset by power-on detection
- Reset by VLS (Voltage Level Supervisor)
- Reset by the 2nd WDT (WatchDog Timer) overflow
- Reset by the WDT (WatchDog Timer) invalid clear reset
- Reset by RAM parity error
- Reset by ROM unused area access
- CPU reset by executing the BRK instruction
- Reset of peripheral circuits by block reset control registers (BRECON0 to 3)

Enabling or disabling the “Reset by ROM unused area access” and “Reset by RAM parity error” is selectable. See Chapter 26 “Code Option” for details about the ROM unused area access reset and See Chapter 29”Safety Function” for the RAM parity error reset”.

Table 3-1 List of Reset Operations for Each Reset Cause

Reset causes	CPU	RAM	VLS0	Peripheral circuit/SFR	Hardware(*)
RESET_N pin	○	-	○	○	○
Power-on reset	○	-	○	○	○
VLS0	○	-	-	○	○
Second overflow of WDT	○	-	-	○	○
WDT counter invalid clear operation	○	-	-	○	○
RAM parity error	○	-	-	○	○
ROM unused area access	○	-	-	○	○
BRK instruction	○	-	-	-	-
Block reset control register	-	-	-	○	-

○: Reset -: Not Reset
(*) Power circuit, oscillation circuit, wake up control circuit and etc.

[Note]
- The system resets do not initialize data memory(RAM) and undefined SFRs. Please initialize them by the software.
- The BRK instruction resets only CPU and does not reset peripheral circuit and other hardwares. Use RESET_N pin reset or WDT reset for surely initializing the LSI in case an unexpected error is detected.
- The block reset control registers (BRECON0 to 3) reset only the peripheral circuits and the CPU and any other hardware are not initialized, also the LSI does not enter the system reset mode.
- Internal pull-up resistor is not installed. Have the pull-up register externally.

3.3.2 Operation of System Reset Mode

The MCU enters to the system reset mode if any reset factors except for block control register(BRECON0-3) occurred.
The system reset has the highest priority among all the processing and any other processing being executed up to then is cancelled. In system reset mode, the following processing is performed.

(1) The hardware such as power supply circuit and oscillation circuit are initialized, and functions selected by the Code Option are configured. INITE bit of Reset Status register (RSTAT) is set to "1" if the initialization failed by any unexpected cause. See the Chapter 26"Code Option" for details about the Code Option.
(2) All the special function registers (SFRs) whose initial value is not undefined are initialized. See Appendix A "Registers" for the initial values of the SFRs.
(3) CPU is initialized.

- All the registers in CPU are initialized.
- The contents of addresses 0x0000 and 0x0001 in the program memory are set to the stack pointer (SP).
- The contents of addresses 0x0002 and 0x0003 in the program memory are set to the program counter (PC). However, in the case of CPU reset by BRK instruction, the address set to the PC is different depending on the value of the interrupt level (ELEVEL) of the program status word (PSW).
  - When the ELEVEL is 0 or 1, the addresses 0x0004 and 0x0005 (segment0) of the program memory are set to the PC.
  - When the ELEVEL is 2 or 3, the addresses 0x0002 and 0x0003 (segment0) of the program memory are set to the PC.

(4) The MCU enters to the program run mode if the reset released.

For the BRK instruction, see "nX-U16/100 Core Instruction Manual".

3.3.3 RESET_N pin

The MCU enters to the system reset mode if any reset factors except for block control register(BRECON0-3) occurred.
Asserting "L" level into the RESET_N pin makes the MCU reset state and RSTR bit of Reset Status Register(RSTAT) is set to "1". Then, when it turned to "H" level the reset state gets released and the CPU starts runing the program code.
Remain the "L" level for the time of reset hold time (PRST) or longer.
Figure3-2 shows the reset signal input waveform.

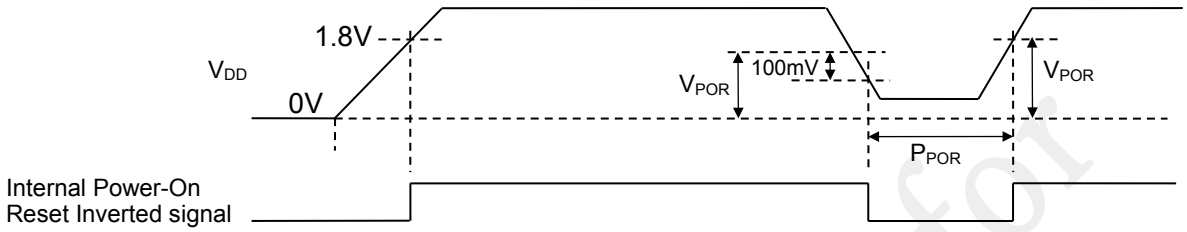
Figure3-2 RESET_N pin reset operating waveform

The RESET_N pin is also used in debugging or programming the software by the on-chip debugger or ISP function. See Chapter 28 "On-chip debug function" and Chapter 25 "Flash Memory".
For details about the electrical characteristics, see the data sheet.

[Note]
- Internal pull-up resistor is not installed. Have the pull-up register externally.

3.3.4 Power-On Reset

The Power-On Reset occurs when it detects the POR detect voltage(V_{POR}) or lower and POR bit of Reset Status Register (RSTAT) is set to "1". Then when the power V_{DD} gets back to the V_{POR}, the reset is release and the CPU starts running the program code. For details about the electrical characteristics, see the data sheet.



V_{POR}: POR detect voltage
P_{POR}: POR response time
(This is the time from the V_{DD} gets 100mV lower than V_{POR} to the Power-On-Reset internally generates)

Figure3-3 Power-On reset operating waveform

[Note]
- Rise the VDD up to 1.8V or higher at the power up.
- At the power-on, the reset is released when the power voltage(VDD) is higher than the power-on reset detect voltage(VPOR) and the CPU starts running the program with low-speed clock(LSCLK/approx. 32.768kHz). Make one of following process when switching the CPU clock to a high-speed clock.
  - Remain reset by the VLS function as long as the VDD is lower than the operate-able voltage (1.8V) for the high-speed clock.
  - Check with the VLS function if the VDD is higher than operate-able voltage (1.8V) to switch the clock.

  See For details about the VLS, see Chapter 22 " Voltage Level Supervisor (VLS)".
- At the power-off, the reset occurs when the power voltage(VDD) is lower than the power-on reset detect voltage(VPOR), however make the VLS reset before the VDD is lower than the minimum operating voltage described in the electrical characteristics of data sheet. Also, confirm if the voltage has returned within the operating voltage when the CPU restarts running the program.

Chapter 4 Power Management

4. Power Management

4.1 General Description

ML62Q1000 series has four power management modes to save the current consumption.
See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

- HALT mode :Stop the CPU and peripherals continue to work.
- HALT-H mode :Stop the CPU, peripherals continue to work with low-speed clock only, forcely stop high-speed clock and restart the high-speed clcok after releasing the mode.
- STOP mode :Stop the CPU, peripheral circuits, low-speed clcok and high-speed clock.
- STOP-D mode : Stop the CPU, peripheral circuits, low-speed clcok and high-speed clock. V_{DDL} is minimised to lower the current consumption.

4.1.1 Features

- Stop code accceptor qualifies for entering STOP mode and STOP-D mode
- Data of RAM and SFR are retained even in the STOP-D mode
- 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 4-1 shows the transition diagram of the operating state.

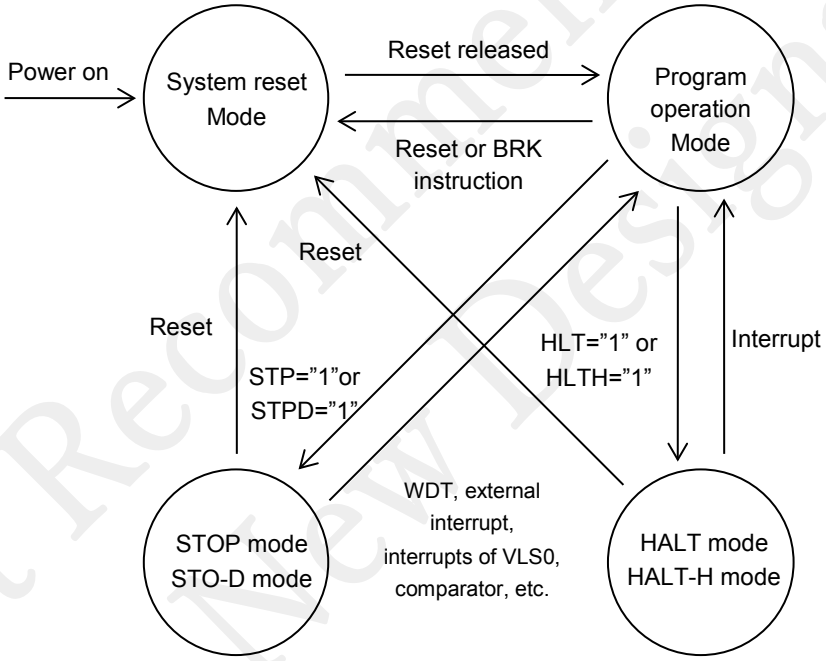


Figure 4-1 Operating State Transition Diagram

4.2 Description of Registers

4.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF018	Stop code acceptor	STPACP	-	W	8	-
0xF019	Reserved register	-	-	W	8	0x00
0xF01A	Standby control register L	SBYCONL	SBYCON	W	8/16	0x00
0xF01B	Standby control register H	SBYCONH	-	W	8	0x00
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

[Note]
For registers with word symbol, word access is possible. For word access, specify an even address.

4.2.2 Stop Code Acceptor (STPACP)

Address: 0xF018
Access: W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								STPACP							
Bit symbol	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
Access type	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

STPACP is a write-only specific function register (SFR) to be used to change the operating state into the STOP mode and STOP-D mode.
When STPACP is read, "0x00" is read.
When values "0x5n" and "0xAn" (n = 0 to 0F) are written in STPACP in this order, it allows once entering to the STOP mode or STOP-D. In this state, when "1" is written in STP or STPD bit in the standby control register (SBYCON), the operating state enters the STOP mode or STOP-D 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. This stop code acceptor gets invalid by the system reset.

[Note]
Writing to the stop code acceptor is invalid on the condition both interrupt enable bits and interrupt request bits are “1, it will not get enabled for entering to the STOP mode and STOP-D mode.

4.2.3 Standby Control Register (SBYCON)

Address: 0xF01A
Access: W
Access size: 8/16 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SBYCON															
Byte symbol	SBYCONH								SBYCONL							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	STPD	HLTH	STP	HLT
Access type	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

SBYCON is a specific function register (SFR) to control the power management.

Description of bits

- **HLT** (Bit 0)
  HLT is a bit to change the operating state into the HALT mode.
  When the WDT interrupt or an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the HALT mode gets canceled and returns to program run mode.

- **STP** (Bit 1)
  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.
  When the WDT interrupt or an external interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the STOP mode gets canceled and returns to program run mode.

- **HLTH** (Bit 2)
  HLTH is a bit to stop forcily the high-speed oscillation and change the operating state into the HALT-H mode.
  When the WDT interrupt or an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the HLTH mode gets canceled and returns to program run mode after enabling the high-speed oscillations forcibly.

- **STPD** (Bit 3)
  STPD is a bit to change the operating state into the STOP-D mode. When "1" is written in the STPD bit after entering the STOP mode is allowed by using STPACP, the operating state enters the STOP-D mode.
  When the WDT interrupt or an external interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the STOP-D mode gets canceled and returns to program run mode.

[Note]
- The operating state does not enter the standby mode under the condition that both an interrupt enable flag and an interrupt request flag are “1” that is requesting the interrupt to the CPU.
- When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condtion of MIE flag of the program status word (PSW) is "0", it cancels the stnadby mode only and does not enter the interrupt processing. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".
- Insert two NOP instructions after the instruction of that sets HLT, STP, HLTH and STPD bit to “1”. The operation without the two NOP instructions is not guaranteed. When using the automatic CRC calculation mode, See the program example described in Chapter 19.3.3. “Automatic CRC Calculation Mode”.
- If two bits or more in the SBYCON are set to “1” at the same time, the setting are gets invalid and continues the program rum mode.

4.2.4 Block Clock Control Register 0 (BCKCON0)

Address: 0xF070
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BCKCON0															
Byte symbol	BCKCON0H								BCKCON0L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	DCKTM5	DCKTM4	DCKTM3	DCKTM2	DCKTM1	DCKTM0
Access type	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

BCKCON0 is a specific function register (SFR) to enable or disable the clock supply (high-speed/low-speed) to each block.
Power consumption can be reduced by stopping clock supply of unused peripheral circuits.
The bits are unwriteable when the products do not have the peripheral function and always return “0” for reading.

Description of setting value

Setting value	Description
0	Clock supply to corresponding peripheral circuits enabled (initial value)
1	Clock supply to corresponding peripheral circuits disabled

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	DCKTM0	16-bit general-purpose timer 0
Bit 1	DCKTM1	16-bit general-purpose timer 1
Bit 2	DCKTM2	16-bit general-purpose timer 2
Bit 3	DCKTM3	16-bit general-purpose timer 3
Bit 4	DCKTM4	16-bit general-purpose timer 4
Bit 5	DCKTM5	16-bit general-purpose timer 5
Bit 6	−	−
Bit 7	−	−
Bit 8	−	−
Bit 9	−	−
Bit 10	−	−
Bit 11	−	−
Bit 12	−	−
Bit 13	−	−
Bit 14	−	−
Bit 15	−	−

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.5 Block Clock Control Register 1 (BCKCON1)

Address: 0xF072
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BCKCON1															
Byte symbol	BCKCON1H								BCKCON1L							
Bit symbol	-	-	-	DCKI2CU0	-	-	-	DCKI2CM0	-	-	-	-	DCKFTM3	DCKFTM2	DCKFTM1	DCKFTM0
Access type	R	R	R	R/W	R	R	R	R/W	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

BCKCON1 is a specific function register (SFR) to enable or disable clock (high-speed/low speed) supply to each block. The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Clock supply to corresponding peripheral circuits enabled (initial value)
1	Clock supply to corresponding peripheral circuits disabled

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	DCKFTM0	Functional timer 0
Bit 1	DCKFTM1	Functional timer 1
Bit 2	DCKFTM2	Functional timer 2
Bit 3	DCKFTM3	Functional timer 3
Bit 4	-	-
Bit 5	-	-
Bit 6	-	-
Bit 7	-	-
Bit 8	DCKI2CM0	I^2C master 0
Bit 9	-	-
Bit 10	-	-
Bit 11	-	-
Bit 12	DCKI2CU0	I^2C bus unit 0
Bit 13	-	-
Bit 14	-	-
Bit 15	-	-

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.6 Block Clock Control Register 2 (BCKCON2)

Address: 0xF074
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BCKCON2															
Byte symbol	BCKCON2H								BCKCON2L							
Bit symbol	DCKDMA	DCKBUZ	DCKACC	-	DCKCRC	-	-	-	-	-	-	-	-	-	DCKSU1	DCKSU0
Access type	R/W	R/W	R/W	R	R/W	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

BCKCON2 is a specific function register (SFR) to enable or disable clock (high-speed/low speed) supply to each block. The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Clock supply to corresponding peripheral circuits enabled (initial value)
1	Clock supply to corresponding peripheral circuits disabled

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	DCKSU0	Serial unit communication function 0
Bit 1	DCKSU1	Serial unit communication function 1
Bit 2	-	-
Bit 3	-	-
Bit 4	-	-
Bit 5	-	-
Bit 6	-	-
Bit 7	-	-
Bit 8	-	-
Bit 9	-	-
Bit 10	-	-
Bit 11	DCKCRC	CRC operation circuit
Bit 12	-	-
Bit 13	DCKACC	Multiplication/division circuit
Bit 14	DCKBUZ	Buzzer
Bit 15	DCKDMA	DMAC

[Note]
- DCKACC can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified in the target option of the integrated development environment IDEU8.
- To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.7 Block Clock Control Register 3 (BCKCON3)

Address: 0xF076
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BCKCON3															
Byte symbol	BCKCON3H								BCKCON3L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	DCKCMP0	-	-	DCKDAC	DCKSAD
Access type	R	R	R	R	R	R	R	R	R	R/W	R	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

BCKCON3 is a specific function register (SFR) to enable or disable clock (high-speed/low speed) supply to each block.
Bit6 is reserved, always write "0".
The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Clock supply to corresponding peripheral circuits enabled (initial value)
1	Clock supply to corresponding peripheral circuits disabled

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	DCKSAD	Successive approximation type A/D converter
Bit 1	DCKDAC	D/A converter
Bit 2	-	-
Bit 3	-	-
Bit 4	DCKCMP0	Comparator 0
Bit 5	-	-
Bit 6	rsvd	Reserved bit
Bit 7	-	-
Bit 8	-	-
Bit 9	-	-
Bit 10	-	-
Bit 11		
Bit 12	-	-
Bit 13	-	-
Bit 14	-	-
Bit 15	-	-

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.8 Block Reset Control Register 0 (BRECON0)

Address: 0xF078
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BRECON0															
Byte symbol	BRECON0H								BRECON0L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	RSETM5	RSETM4	RSETM3	RSETM2	RSETM1	RSETM0
Access type	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

BRECON0 is a specific function register (SFR) to control a reset signal to each block.
The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Do not reset the peripheral circuit (initial value)
1	Remain to reset the peripheral circuit

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	RSETM0	16-bit general-purpose timer 0
Bit 1	RSETM1	16-bit general-purpose timer 1
Bit 2	RSETM2	16-bit general-purpose timer 2
Bit 3	RSETM3	16-bit general-purpose timer 3
Bit 4	RSETM4	16-bit general-purpose timer 4
Bit 5	RSETM5	16-bit general-purpose timer 5
Bit 6	−	−
Bit 7	−	−
Bit 8	−	−
Bit 9	−	−
Bit 10	−	−
Bit 11	−	−
Bit 12	−	−
Bit 13	−	−
Bit 14	−	−
Bit 15	−	−

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.9 Block Reset Control Register 1 (BRECON1)

Address: 0xF07A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BRECON1															
Byte symbol	BRECON1H								BRECON1L							
Bit symbol	-	-	-	RSEI2CU0	-	-	-	RSEI2CM0	-	-	-	-	RSEFTM3	RSEFTM2	RSEFTM1	RSEFTM0
Access type	R	R	R	R/W	R	R	R	R/W	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

BRECON1 is a specific function register (SFR) to control a reset signal to each block.
The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Do not reset the peripheral circuit (initial value)
1	Remain to reset the peripheral circuit

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	RSEFTM0	Functional timer 0
Bit 1	RSEFTM1	Functional timer 1
Bit 2	RSEFTM2	Functional timer 2
Bit 3	RSEFTM3	Functional timer 3
Bit 4	-	-
Bit 5	-	-
Bit 6	-	-
Bit 7		
Bit 8	RSEI2CM0	I^2C master 0
Bit 9	-	-
Bit 10	-	-
Bit 11	-	-
Bit 12	RSEI2CU0	I^2C bus unit 0
Bit 13	-	-
Bit 14	-	-
Bit 15	-	-

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.10 Block Reset Control Register 2 (BRECON2)

Address: 0xF07C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BRECON2															
Byte symbol	BRECON2H								BRECON2L							
Bit symbol	RSEDMA	RSEBUZ	RSEACC	-	RSECRC	-	-	-	-	-	-	-	-	-	RSESU1	RSESU0
Access type	R/W	R/W	R/W	R	R/W	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

BRECON2 is a specific function register (SFR) to control a reset signal to each block.
The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Do not reset the peripheral circuit (initial value)
1	Remain to reset the peripheral circuit

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	RSESU0	Serial unit communication function 0
Bit 1	RSESU1	Serial unit communication function 1
Bit 2	-	-
Bit 3	-	-
Bit 4	-	-
Bit 5	-	-
Bit 6	-	-
Bit 7	-	-
Bit 8	-	-
Bit 9	-	-
Bit 10	-	-
Bit 11	RSECRC	CRC operation circuit
Bit 12	-	-
Bit 13	RSEACC	Multiplication/division circuit
Bit 14	RSEBUZ	Buzzer
Bit 15	RSEDMA	DMAC

[Note]
- RSEACC bit is can be set to "1" when not specifying "muldivu8.lib" in [Target Option] of IDEU8 integrated software development environment.
- To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

4.2.11 Block Reset Control Register 3 (BRECON3)

Address: 0xF07E
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BRECON3															
Byte symbol	BRECON3H								BRECON3L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	RRECMP0	-	-	RREDAC	RRESAD
Access type	R	R	R	R	R	R	R	R	R	R	R	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

BRECON3 is a specific function register (SFR) to control a reset signal to each block.
The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading.

Description of setting value

Setting value	Description
0	Do not reset the peripheral circuit (initial value)
1	Remain to reset the peripheral circuit

[Note] POR function turns OFF when "1" is set to the POR bit.

Corresponding block

Bit	Bit symbol name	Corresponding block
Bit 0	RRESAD	Successive approximation type A/D converter
Bit 1	RREDAC	D/A converter
Bit 2	-	-
Bit 3	-	-
Bit 4	RRECMP0	Comparator 0
Bit 5	-	-
Bit 6	-	-
Bit 7	-	-
Bit 8	-	-
Bit 9	-	-
Bit 10	-	-
Bit 11	-	-
Bit 12	-	-
Bit 13	-	-
Bit 14	-	-
Bit 15	-	-

[Note]
To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

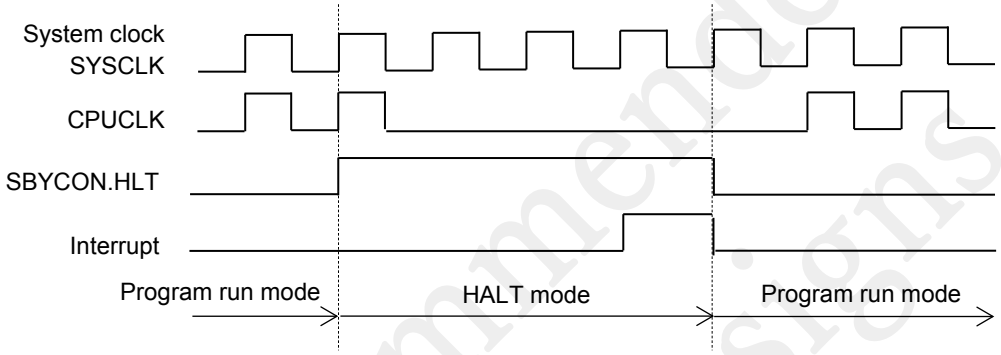
4.3 Description of Operation

4.3.1 Program Run Mode

The program run mode is the state the CPU executes instructions sequentially.
The state changes from the system reset mode to the program run mode after a reset gets occurred and released.
A standby mode (HALT mode, HALT-H mode, STOP mode and STOP-D mode) is waken up by an interrupt request and returned to the program run mode.
For details about the system reset mode, see Chapter 3 “Reset Function”.

4.3.2 HALT Mode

The HALT mode is the state the CPU stops executing instructions remaining the system clock selected before entering the HALT mode and only the peripheral circuits are working. See section 4.3.7 “Operation of Functions in the standby modes” for operation of peripheral functions in the HALT mode.
When "1" is written in the HLT bit in the standby control register (SBYCON), the operating state enters the HALT mode.
When a WDT interrupt or an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the HALT mode is released at the rising edge of the next system clock(SYSCLK) and returns to the program run mode with the previous system clock.
Figure 4-2 shows the operation waveforms in the HALT mode.



SBYCON：Standby control part register

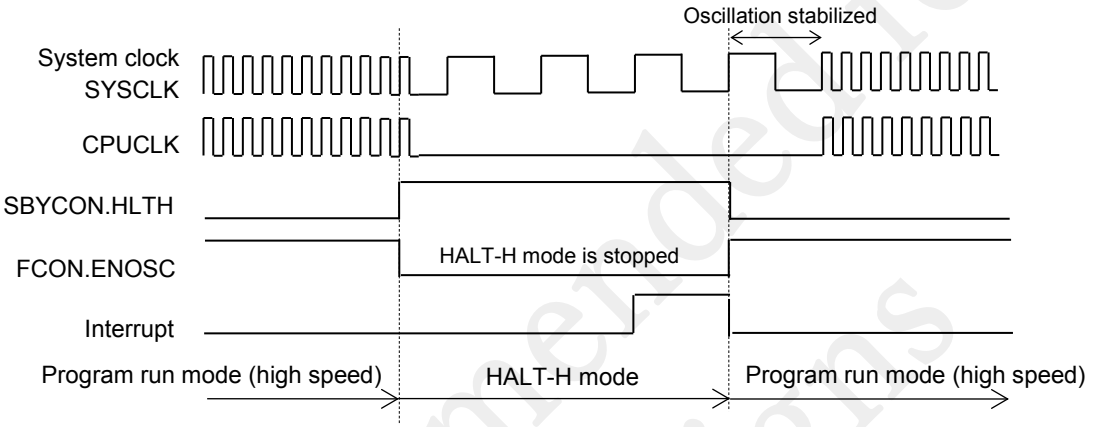
Figure 4-2 Operation Waveforms in HALT Mode

4.3.3 HALT-H mode

The HALT-H mode is the state the high-speed oscillation is stopped, the CPU stops executing instructions and only the peripheral circuits are working. The periperal circuits that worked with the high-speed clock gets stopped. See section 4.3.7 “Operation of Functions in the standby modes” for operation of peripheral functions in the HALT-H mode.

When "1" is written to the HLTH bit in the standby control register (SBYCON), the operating state enters the HALT-H mode. When a WDT interrupt or an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated, the HALT-H mode is released at the rising edge of the next system clock(SYSCLK) and the high-speed oscillation forcibly gets enabled and returns to the program run mode with the high-speed system clock. Even if the high-speed oscillation was disabled(ENOSC bit =”0”) and the low-speed clock LSCLK was seleted (SELSCLK bit = “0”), the high-speed oscilation and the high-speed system clock HSCLK was forcibly seleteced after releasing the HALT-H mode (i.e. forcibly ENOSC bit gets “1” and SELSCLK gets “1”).

Figure 4-3 shows the operation waveforms in the high-speed HALT mode.



SBYCON : Standby control part register
FCON : Clock control register

Figure 4-3 Operation Waveforms in HALT-H Mode

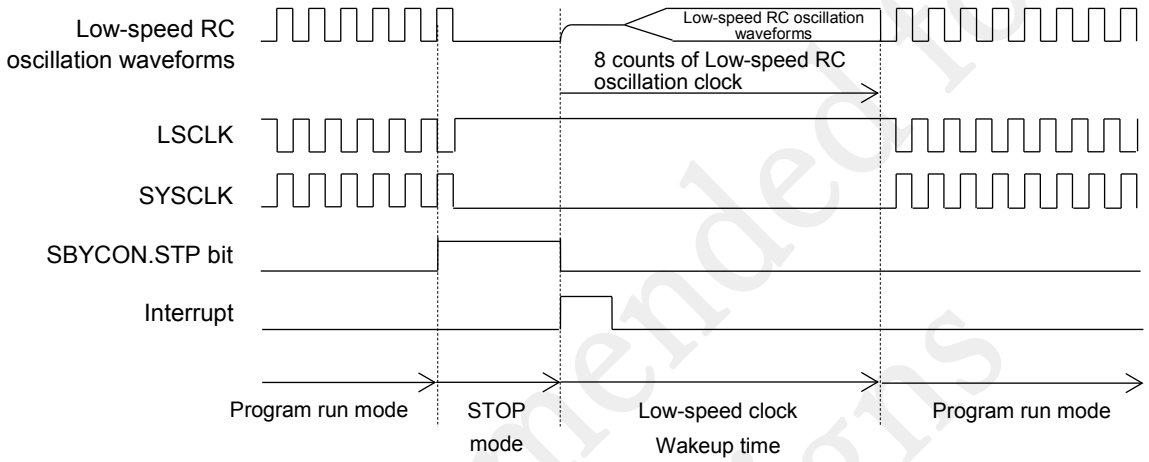
4.3.4 STOP mode

The STOP mode is the state both the low-speed osciallation clock and the high-speed oscillation clock are forcibly stopped, the CPU stops executing instructions and the peripheral circuits which need the clock stop working. See section 4.3.7 “Operation of Functions in the standby modes” for operation of peripheral functions in the STOP mode.

To enter the STOP mode, write "5nH" then "0AnH" (n = 0 to 0FH) into the stop code acceptor (STPACP) to enable the STPACP, and write "1" into the STP bit of the standby control register (SBYCON).

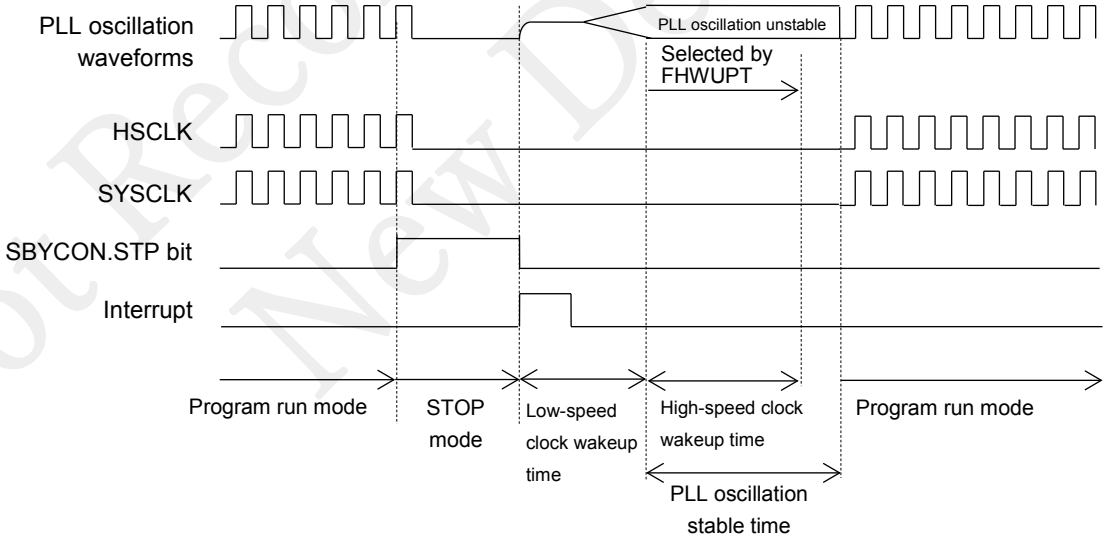
The STOP mode is released by the external interrupts, comparator interrupts or the watch dog timer(WDT) interrupts and returns to the program run mode with the system clock previously selected.

Figure 4-4 shows the STOP mode operation waveforms of the low-speed RC oscillation circuit.
Figure 4-5 shows the STOP mode operation waveforms of the PLL oscillation circuit.



SBYCON : Standby control part register

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



SBYCON : Standby control part register
FHWUPT : High-speed clock wakeup time setting register
For details of FHWUPT, see Chapter 6 "Clock Generation Circuit."

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

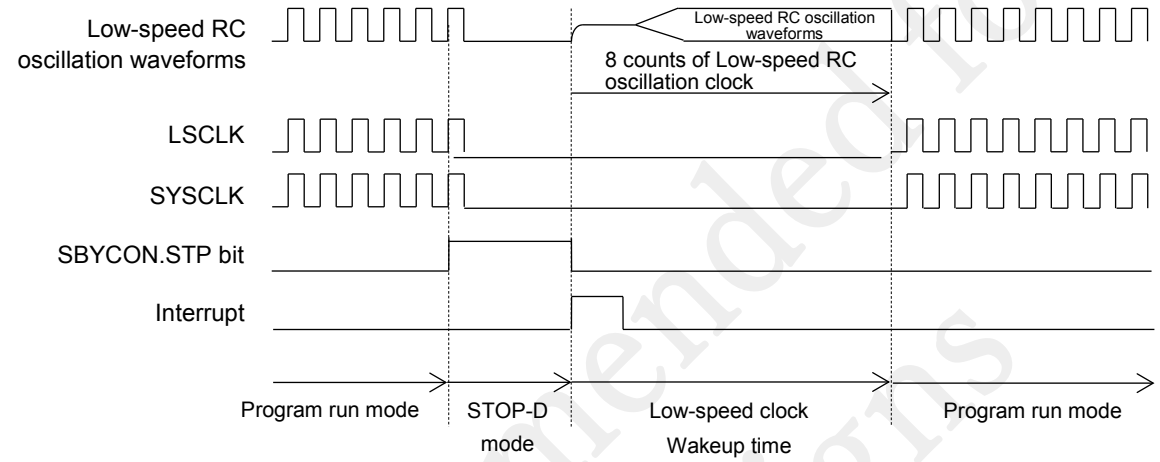
4.3.5 STOP-D Mode

The STOP-D mode is that an internal voltage conrol is added to the STOP mode. It can reduce the current consumption by making lower the internal logic power supply(V_{DDL}). See section 4.3.7 “Operation of Functions in the standby modes” for operation of peripheral functions in the STOP-D mode.

To enter the STOP-D mode, write "5nH" then "0AnH" (n = 0 to 0FH) into the stop code acceptor (STPACP) to enable the STPACP, and write "1" into the STPD bit of the standby control register (SBYCON).

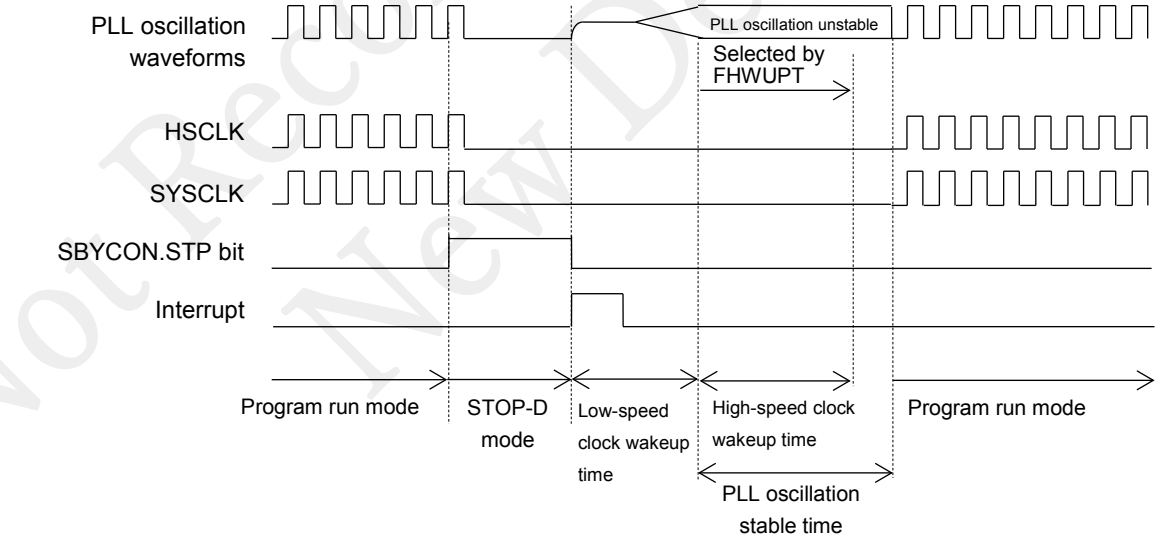
The STOP-D mode is released by the external interrupts, comparator interrupts or the watch dog timer(WDT) interrupts and returns to the program run mode with the system clock previously selected.

Figure 4-6 shows the STOP-D mode operation waveforms of the low-speed RC oscillation circuit.
Figure 4-7 shows the STOP-D mode operation waveforms of the PLL oscillation circuit.



SBYCON : Standby control part register

Figure 4-6 STOP-D Mode Operation Waveforms of Low-Speed RC Oscillation Circuit



SBYCON : Standby control part register
FHWUPT : High-speed clock wakeup time setting register
For details of FHWUPT, see Chapter 6 "Clock Generation Circuit."

Figure 4-7 STOP-D Mode Operation Waveforms of PLL Oscillation Circuit

4.3.6 Note on Return Operation from the standby modes

The operation of returning from the STOP/STOP-D mode and the HALT/ HALT-H mode varies according to the interrupt level (ELEVEL) of the program status word (PSW), master interrupt enable flag (MIE), and the contents of the interrupt enable register (IE0 to IE3), as well as whether the interrupt is non-maskable or maskable.
For details of PSW and the IE and IRQ registers, see "nX-U16/100 Core Instruction Manual" and Chapter 5 "Interrupts", respectively.
Tables 4-1 and 4-2 show the return operations from the standy modes.

Table 4-1 Return Operation from the standby modes (for Non-Maskable Interrupt)

ELEVEL	MIE	IEn.m	IRQn.m	Return operation from STOP/HALT/STOP-D/HALT-H mode
*	*	-	0	Not returned from SSTOP/HALT/STOP-D/HALT-H mode
3	*	-	1	After the mode is returned from the STOP/HALT/STOP-D/HALT-H mode, the program operation restarts from the instruction following the instruction that sets the STP/HLT/STPD/HLTH bit to "1." The program operation does not go to the interrupt routine.
0,1,2	*	-	1	After the mode is returned from the STOP/HALT/STOP-D/HALT-H mode, program operation restarts from the instruction following the instruction that sets the STP/HLT/STPD/HLTH bit to "1", then goes to the interrupt routine.

Table 4-2 Return Operation from standby modes (for Maskable Interrupt)

ELEVEL	MIE	IEn.m	IRQn.m	Return operation from STOP/HALT/STOP-D/HALT-H mode
*	*	*	0	Not returned from STOP/HALT/STOP-D /HALT-H mode.
*	*	0	1	
*	0	1	1	After the mode is returned from the STOP/HALT/STOP-D/HALT-H mode, the program operation restarts from the instruction following the instruction that sets the STP/HLT/STPD/HLTH bit to "1." The program operation does not go to the interrupt routine.
2,3	1	1	1	
0,1	1	1	1	After the mode is returned from the STOP/HALT/STOP-D/HALT-H mode, program operation restarts from the instruction following the instruction that sets the STP/HLT/STPD/HLTH bit to "1", then goes to the interrupt routine.

n=0~7, m=0~7
*: Don't care
The interrupt level (ELEVEL) of program status word (PSW) are bits that indicate the state of interrupt process performed by the CPU, which are set by the hardware when transferring to the interrupt process or returning from the interrupt.

- If the ELEVEL bit is 0H, it indicates that the CPU is performing neither non-maskable interrupt processing nor maskable interrupt processing nor software interrupt processing.
- If the ELEVEL bit is 1H, it indicates that the CPU is performing maskable interrupt processing or software interrupt processing. (ELEVEL is set during interrupt transition cycle)
- If the ELEVEL bit is 2H, it indicates that the CPU is performing non-maskable interrupt processing. (ELEVEL is set during interrupt transition cycle)
- If the ELEVEL bit is 3H, it indicates that the CPU is performing interrupt processing specific to the emulator. Do not use this setting in normal applications.

[Note]
- Insert two NOP instructions after the standby mode (HALT, STOP, HALT-H, STOP-D mode) transition instructions, because the maximum two instructions will be executed after releasing the standby mode and before transferring to the interrupt. When the master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", instructions will be executed after the two NOP execution and the interrupt transferring cycles. When the MIE is "0" the CPU executes the two NOP instructions and continues executing the next instructions without transferring to the interrupt.

4.3.7 Operation of Functions in the standby modes

Table 4-3 shows the state of each function in the HALT, HALT-H, STOP and STOP-D modes.

Table 4-3 State of Functions in STOP/HALT/STOP-D/HALT-H Mode

Function	HALT	HALT-H *3	STOP	STOP-D
CPU	-	-	-	-
RAM	Retain	Retain	Retain	Retain
Watchdog timer	○	○	-	-
External interrupt	○	○	○*2	○*2
Time base counter	○	○	-	-
16-bit timer	○	○	-	-
16-bit functional timer	○	○	-	-
Serial unit	○	○	-	-
I^2C unit	○	○	-	-
I^2C master	○	-	-	-
Buzzer	○	○	-	-
A/D converter	○	○	-	-
D/A converter*4	○	○	○	○
Analog Comparator	○	○	○*2	○*2
Voltage Level Detection (VLS)	○	○	○*2*6	○*2*6
BGO operation*5	○	-	-	-
DMA controller	-	-	-	-
CRC Calculator	○	○	-	-
Multiplier/Divider	○	○	-	-

○: Operable -: Not operable (Stop)

*2: The sampling function is forcibly disabled.
*3: If the peripheral circuits are operated with the high-speed clock, they cannot be operated in the HALT-H mode. To operate the peripheral circuits in the HALT-H mode, select the low-speed clock for the operation clock of the peripheral circuits.
*4: The D/A converter remain the previous state during the standby mode.
*5: The BGO means the operation while erasing or programming the data flash.
*6: Reset function is only available (The interrupt is not available).

Table 4-4 shows the wake-up times from the HALT-H, STOP and STOP-D modes. See Chapter 6 “Clock Generation Circuit “ for details about the high-speed clock wake-up time setting register (FHWUPT).

Table 4-4 Wake-up Times from HALT-H/STOP/STOP-D Mode

Function	Low-speed clock (Low-speed RC oscillation)	High-speed clock (PLL oscillation)	
		FHWUPT = 0x01	FHWUPT = 0x00
HALT-H mode	Continue oscillating the clock	Approx. 30μs	Approx. 2.5ms
STOP mode	Approx. 250μs	Approx. 30μs	Approx. 2.5ms
STOP-D mode	Approx. 250μs	Approx. 30μs	Approx. 2.5ms

[Note]

- If the system clock is switched to the high-speed clcok after the STOP or STOP-D mode is released before the wake-up time is passed, the CPU is hold to run the program because the clock has not been supplied.
- Select the low-speed clock for the operation clock of peripheral circuits when operating the peripheral circuits during the HALT-H mode.
- When the high-speed clock wake-up time setting register (FHWUPT) is “0x00”, the output clock of PLL oscillation is masked for approx.2.5ms. The HSCLK starts to be supplying after waiting the 2.5ms. The SYSTEM clock is also stopped during the 2.5ms.
- When the high-speed clock wake-up time setting register (FHWUPT) is “0x01”, the frequency of PLL oscillation clock gradually increses from approx. 1MHz and will get the correct frequency (16MHz/24MHz/32MHz) set by the code option withing the approx.2.5ms. The accuracy of the frequency described in the data sheet cannot be guranteed during the 2.5ms, however it can be used for the system clock without waiting the 2.5ms.

4.3.8 Block Control Function

ML62Q1000 series have the block clock control function that stops the clock to unused peripherals and the block reset control function that resets peripherals. When setting bits of the block clock control registers (BCKCONn, n=0 to 3), it stops supplying the clock to the corresponding peripherals and reduce the current consumption. When setting bits of the block reset control registers (BRECONn, n=0 to 3), it resets the corresponding peripherals and the SFRs get initialized.

Table 4-5 shows the control registers and the corresponding peripherals.

Table 4-5 List of peripherals and control registers

Peripherals	Block clock control function		Block reset control function	
	SFR Word symbol	SFR Bit symbol	SFR Word symbol	SFR Bit symbol
16bit Timer 0	BCKCON0	DCKTM0	BRECON0	RSETM0
16bit Timer 1		DCKTM1		RSETM1
16bit Timer 2		DCKTM2		RSETM2
16bit Timer 3		DCKTM3		RSETM3
16bit Timer 4		DCKTM4		RSETM4
16bit Timer 5		DCKTM5		RSETM5
Functional timer 0	BCKCON1	DCKFTM0	BRECON1	RSEFTM0
Functional timer 1		DCKFTM1		RSEFTM1
Functional timer 2		DCKFTM2		RSEFTM2
Functional timer 3		DCKFTM3		RSEFTM3
I^2C bus master 0		DCKI2CM0		RSEI2CM0
I^2C bus unit 0		DCKI2CU0		RSEI2CU0
Serial communication unit 0	BCKCON2	DCKSU0	BRECON2	RSESU0
Serial communication unit 1		DCKSU1		RSESU1
CRC operation		DCKCRC		RSECRC
Multiplier/Divider		DCKACC		RSEACC
Buzzer		DCKBUZ		RSEBUZ
DMA controller		DCKDMA		RSEDMA
Sucessive A/D converter	BCKCON3	DCKSAD	BRECON3	RSEADC
D/A conveter		DCKDAC		RSEDAC
Analog comparator		DCKCMP0		RSECMP0

The peripherals are enalbed to use after the sytem reset released (the clock is supplied and the reset is released).
For unused peripheral circuits, at first reset the peripheral by setting the corresponding bits of BRECONn(n=0 to 3) to “1” at first and the next, stop the clock by setting the corresponding bits of BCKCONn(n=0 to 3) .
If using the peripheral circuits, at first supply the clock to the periperals by clearing the bits of BCKCONn(n=0 to 3) and the next, release the reset to the peripherals by clearing the bits of BRECONn(n=0 to 3).
Also, the peripheral circuits can be initialized by reseting them remaing to supply the clock.

Writing the SFRs is invalid on the condition of that clock is not supplied to the peripheral or reset the peripheral. The default value is returned for reading.

[Note]
- If the clock supply is stopped to the peripheral circuits without reseting them by the BRECONn(n=0 to 3), it is possible that output levels of timer, serial communications and buzzer are fixed and it makes a large current. Also, it is possible the sucessive A/D conveter, D/A converter and analog comparator may stop on the condition of the large current is flowing. Thefore, please make sure to stop the clock after reseting the periperals by using the block reset control register BRECONn(n=0 to 3) register.

Chapter 5 Interrupts

5. Interrupts

5.1 General Description

ML62Q1000 series has the non-maskable interrupt, maskable interrupts, and a software interrupt (SWI).
For details of each interrupt, see the corresponding chapters.
See “Table 1-2 Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

5.1.1 Features

- Non-maskable interrupt source: 1 (Internal sources: 1)
- Maskable interrupt sources: 28 (Internal sources: 20, External sources: 8)
- Software interrupt (SWI): maximum 64 sources
- Master Interrupt Enable(MIE) flag enables or disables the all maskable interrupts in a lump. See "nX-U16/100 Core Instruction Manual" for more details about MIE.
- Each maskable interrupt has the enable flag in the register IE0~IE7.
- The occurrence of interrupt request is confirmable by checking the request flag in IRQ registers.
- The occurrence of interrupt is makable by setting each request flag by the software in IRQ registers.
- Four four interrupt levels (Level 1 (low) to Level 4 (high)) are available for each maskable interrupt.

5.2 Description of Registers

5.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
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	-	-	R	8	0x00
0xF032	Current interrupt level Management register	CIL	-	R/W	8	0x00
0xF033	Reserved register	-	-	R	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
0F03EH	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

[Note]
For registers with word symbol, word access is possible. For word access, specify an even address.

5.2.2 Interrupt Enable Register 01 (IE01)

Address: 0xF020
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IE01															
Byte symbol	IE1								IE0							
Bit symbol	EPI7	EPI6	EPI5	EPI4	EPI3	EPI2	EPI1	EPI0	-	EVLS0	-	-	-	-	-	-
Access type	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

IE01 is a specific function register (SFR) to enable or disable the interrupt for each interrupt request.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU becomes "0", however, the applicable flag of IE01 is not reset.

Description of setting value

Setting value	Description
0	Interrupt disabled (initial value)
1	Interrupt enabled

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	EPI7	Pin interrupt 7 (EXI7INT)
Bit 14	EPI6	Pin interrupt 6 (EXI6INT)
Bit 13	EPI5	Pin interrupt 5 (EXI5INT)
Bit 12	EPI4	Pin interrupt 4 (EXI4INT)
Bit 11	EPI3	Pin interrupt 3 (EXI3INT)
Bit 10	EPI2	Pin interrupt 2 (EXI2INT)
Bit 9	EPI1	Pin interrupt 1 (EXI1INT)
Bit 8	EPI0	Pin interrupt 0 (EXI0INT)
Bit 7	-	-
Bit 6	EVLS0	VLS0 interrupt (VLS0INT)
Bit 5	-	-
Bit 4	-	-
Bit 3	-	-
Bit 2	-	-
Bit 1	-	-
Bit 0	-	-

5.2.3 Interrupt Enable Register 23 (IE23)

Address: 0xF022
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IE23															
Byte symbol	IE3								IE2							
Bit symbol	ETM1	ETM0	EFTM1	EFTM0	-	EI2CM0	-	-	-	ESAD	-	ESIU01	ESIU00	-	EDMA	-
Access type	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

IE23 is a specific function register (SFR) to disable or enable the interrupt for each interrupt request.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU becomes "0", however, the applicable flag of IE23 is not reset.

Description of setting value

Setting value	Description
0	Interrupt disabled (initial value)
1	Interrupt enabled

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	ETM1	16-bit timer 1 interrupt (TM1INT)
Bit 14	ETM0	16-bit timer 0 interrupt (TM0INT)
Bit 13	EFTM1	Functional timer 1 interrupt (FTM1INT)
Bit 12	EFTM0	Functional timer 0 interrupt (FTM0INT)
Bit 11	-	-
Bit 10	EI2CM0	I2C master 0 interrupt (I2CM0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	-	-
Bit 6	ESAD	Successive approximation type A/D interrupt (SADINT)
Bit 5	-	-
Bit 4	ESIU01	Serial unit 01 interrupt (SIU01INT)
Bit 3	ESIU00	Serial unit 00 interrupt (SIU00INT)
Bit 2	-	-
Bit 1	EDMA	DMAC interrupt (DMACINT)
Bit 0	-	-

5.2.4 Interrupt Enable Register 45 (IE45)

Address: 0xF024
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IE45															
Byte symbol	IE5								IE4							
Bit symbol	ETM5	ETM4	-	-	-	ECMP0	-	-	ETM3	ETM2	EFTM3	EFTM2	-	ESIU11	ESIU10	EI2CU0
Access type	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

IE45 is a specific function register (SFR) to enable or disable the interrupt for each interrupt request.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU becomes "0", however, the applicable flag of IE45 is not reset.

Description of setting value

Setting value	Description
0	Interrupt disabled (initial value)
1	Interrupt enabled

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	ETM5	16-bit timer 5 interrupt (TM5INT)
Bit 14	ETM4	16-bit timer 4 interrupt (TM4INT)
Bit 13	-	-
Bit 12	-	-
Bit 11	-	-
Bit 10	ECMP0	Comparator interrupt (CMP0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	ETM3	16-bit timer 3 interrupt (TM3INT)
Bit 6	ETM2	16-bit timer 2 interrupt (TM2INT)
Bit 5	EFTM3	Functional timer 3 interrupt (FTM3INT)
Bit 4	EFTM2	Functional timer 2 interrupt (FTM2INT)
Bit 3	-	-
Bit 2	ESIU11	Serial unit 11 interrupt (SIU11INT)
Bit 1	ESIU10	Serial unit 10 interrupt (SIU10INT)
Bit 0	EI2CU0	I2C master/slave 0 interrupt (I2CU0INT)

5.2.5 Interrupt Enable Register 67 (IE67)

Address: 0xF026
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IE67															
Byte symbol	IE7								IE6							
Bit symbol	-	-	ELTBC2	ELTBC1	-	ELTBC0	-	-	-	-	-	-	-	-	-	-
Access type	R	R	R/W	R/W	R	R/W	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

IE67 is a specific function register (SFR) to enable or disable the interrupt for each interrupt request.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU becomes "0", however, the applicable flag of IE67 is not reset.

Description of setting value

Setting value	Description
0	Interrupt disabled (initial value)
1	Enable interrupt

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	-	-
Bit 14	-	-
Bit 13	ELTBC2	Time base counter 2 interrupt (LTBC2INT)
Bit 12	ELTBC1	Time base counter 1 interrupt (LTBC1INT)
Bit 11	-	-
Bit 10	ELTBC0	Time base counter 0 interrupt (LTBC0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	-	-
Bit 6	-	-
Bit 5	-	-
Bit 4	-	-
Bit 3	-	-
Bit 2	-	-
Bit 1	-	-
Bit 0	-	-

5.2.6 Interrupt Request Register 01 (IRQ01)

Address: 0xF028
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IRQ01															
Byte symbol	IRQ1								IRQ0							
Bit symbol	QPI7	QPI6	QPI5	QPI4	QPI3	QPI2	QPI1	QPI0	-	QVLS0	-	-	-	-	-	QWDT
Access type	R/W	R/W	R/W	R/W	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

IRQ01 is a specific function register (SFR) to request an interrupt for each interrupt.
Each request flag of IRQ01 becomes "1" when an interrupt is generated, regardless of the values of the interrupt enable register (IE01) and master interrupt enable flag (MIE) of the CPU. At that time, an interrupt is requested to the CPU when the applicable flag of IE01 and MIE are "1."
In addition, an interrupt can be generated by writing "1" to the request flag of IRQ01. In this case, the interrupt transition cycle is started after the next one instruction is executed.
The applicable flag of IRQ01 becomes "0" automatically when the interrupt request is accepted by the CPU.

Description of setting value

Setting value	Description
0	Interrupt request not exist (initial value)
1	Interrupt request exists

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	QPI7	Pin interrupt 7 (EXI7INT)
Bit 14	QPI6	Pin interrupt 6 (EXI6INT)
Bit 13	QPI5	Pin interrupt 5 (EXI5INT)
Bit 12	QPI4	Pin interrupt 4 (EXI4INT)
Bit 11	QPI3	Pin interrupt 3 (EXI3INT)
Bit 10	QPI2	Pin interrupt 2 (EXI2INT)
Bit 9	QPI1	Pin interrupt 1 (EXI1INT)
Bit 8	QPI0	Pin interrupt 0 (EXI0INT)
Bit 7	-	-
Bit 6	QVLS0	VLS0 interrupt (VLS0INT)
Bit 5	-	-
Bit 4	-	-
Bit 3	-	-
Bit 2	-	-
Bit 1	-	-
Bit 0	QWDT	WDT interrupt (WDTINT)

5.2.7 Interrupt Request Register 23 (IRQ23)

Address: 0xF02A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IRQ23															
Byte symbol	IRQ3								IRQ2							
Bit symbol	QTM1	QTM0	QFTM1	QFTM0	-	QI2CM0	-	-	-	QSAD	-	QSIU01	QSIU00	-	QDMA	-
Access type	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R/W	R	R/W	R/W	R	R/W	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQ23 is a specific function register (SFR) to request an interrupt for each interrupt.
Each request flag of IRQ23 becomes "1" when an interrupt is generated, regardless of the values of the interrupt enable register (IE23) and master interrupt enable flag (MIE) of the CPU. At that time, an interrupt is requested to the CPU when the applicable flag of IE23 and MIE are "1."
In addition, an interrupt can be generated by writing "1" to the request flag of IRQ23. In this case, the interrupt transition cycle is started after the next one instruction is executed.
The applicable flag of IRQ23 becomes "0" automatically when the interrupt request is accepted by the CPU.

Description of setting value

Setting value	Description
0	Interrupt request not exist (initial value)
1	Interrupt request exists

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	QTM1	16-bit timer 1 interrupt (TM1INT)
Bit 14	QTM0	16-bit timer 0 interrupt (TM0INT)
Bit 13	QFTM1	Functional timer 1 interrupt (FTM1INT)
Bit 12	QFTM0	Functional timer 0 interrupt (FTM0INT)
Bit 11	-	-
Bit 10	QI2CM0	I2C master 0 interrupt (I2CM0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	-	-
Bit 6	QSAD	Successive approximation type A/D interrupt (SADINT)
Bit 5	-	-
Bit 4	QSIU01	Serial unit 01 interrupt (SIU01INT)
Bit 3	QSIU00	Serial unit 00 interrupt (SIU00INT)
Bit 2	-	-
Bit 1	QDMA	DMAC interrupt (DMACINT)
Bit 0	-	-

5.2.8 Interrupt Request Register 45 (IRQ45)

Address: 0xF02C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IRQ45															
Byte symbol	IRQ5								IRQ4							
Bit symbol	QTM5	QTM4	-	-	-	QCMP0	-	-	QTM3	QTM2	QFTM3	QFTM2	-	QSIU11	QSIU10	QI2CU0
Access type	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	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

IRQ45 is a specific function register (SFR) to request an interrupt for each interrupt.
Each request flag of IRQ45 becomes "1" when an interrupt is generated, regardless of the values of the interrupt enable register (IE45) and master interrupt enable flag (MIE) of the CPU. At that time, an interrupt is requested to the CPU when the applicable flag of IE45 and MIE are "1."
In addition, an interrupt can be generated by writing "1" to the request flag of IRQ45. In this case, the interrupt transition cycle is started after the next one instruction is executed.
The applicable flag of IRQ45 becomes "0" automatically when the interrupt request is accepted by the CPU.

Description of setting value

Setting value	Description
0	Interrupt request not exist (initial value)
1	Interrupt request exists

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	ETM5	16-bit timer 5 interrupt (TM5INT)
Bit 14	QTM4	16-bit timer 4 interrupt (TM4INT)
Bit 13	-	-
Bit 12	-	-
Bit 11	-	-
Bit 10	QCMP0	Comparator interrupt (CMP0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	QTM3	16-bit timer 3 interrupt (TM3INT)
Bit 6	QTM2	16-bit timer 2 interrupt (TM2INT)
Bit 5	QFTM3	Functional timer 3 interrupt (FTM3INT)
Bit 4	QFTM2	Functional timer 2 interrupt (FTM2INT)
Bit 3	-	-
Bit 2	QSIU11	Serial unit 11 interrupt (SIU11INT)
Bit 1	QSIU10	Serial unit 10 interrupt (SIU10INT)
Bit 0	QI2CU0	I2C master/slave 0 interrupt (I2CU0INT)

5.2.9 Interrupt Request Register 67 (IRQ67)

Address: 0xF02E
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	IRQ67															
Byte symbol	IRQ7								IRQ6							
Bit symbol	-	-	QLTBC2	QLTBC1	-	QLTBC0	-	-	-	-	-	-	-	-	-	-
Access type	R	R	R/W	R/W	R	R/W	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

IRQ67 is a specific function register (SFR) to request an interrupt for each interrupt.
Each request flag of IRQ67 becomes "1" when an interrupt is generated, regardless of the values of the interrupt enable register (IE67) and master interrupt enable flag (MIE) of the CPU. At that time, an interrupt is requested to the CPU when the applicable flag of IE67 and MIE are "1." In addition, an interrupt can be generated by writing "1" to the request flag of IRQ67.
The applicable flag of IRQ67 becomes "0" automatically when the interrupt request is accepted by the CPU.

Description of setting value

Setting value	Description
0	Interrupt request not exist (initial value)
1	Interrupt request exists

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 15	-	-
Bit 14	-	-
Bit 13	QLTBC2	Time base counter 2 interrupt (LTBC2INT)
Bit 12	QLTBC1	Time base counter 1 interrupt (LTBC1INT)
Bit 11	-	-
Bit 10	QLTBC0	Time base counter 0 interrupt (LTBC0INT)
Bit 9	-	-
Bit 8	-	-
Bit 7	-	-
Bit 6	-	-
Bit 5	-	-
Bit 4	-	-
Bit 3	-	-
Bit 2	-	-
Bit 1	-	-
Bit 0	-	-

5.2.10 Interrupt Level Control Enable Register (ILEN)

Address: 0xF030
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								ILEN							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ILE
Access type	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

ILEN is a specific function register (SFR) to enable or disable the interrupt level control.
When the interrupt level control is disabled, multiple interrupts are output to the CPU according to the priority specified by the hardware.
For details of the priority, see Table 5-1 List of Interrupt Sources in "5.1.1 Features."

Description of bits

- **ILE** (Bit 0)
  ILE bit enables or disables the interrupt level control.

ILE	Description
0	Disabled (initial value)
1	Enabled

[Note]
- Disable the inerrupt level control function by reseting the ILE bit to “0” after resetting the Interrupt level control register ILCn0 and ILCn1(n=0~7) to “0x00” and confirming the current interrupt request level register (CIL) is “0x00” when the interrupt is disabled(IE0~IE7 registers are “0x00”)
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~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)

Address: 0xF032
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CIL							
Bit symbol	-	-	-	-	-	-	-	-	CILN	-	-	-	CILM3	CILM2	CILM1	CILM0
Access type	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

The current interrupt level management register (CIL) is a specific function register (SFR) to manage the priority level of the interrupt currently being processed by the processor.
After maskable or non-maskable interrupts to which the priority levels are specified by the interrupt level control registers (ILC0~7) is accepted by the CPU, corresponding bits of CIL are automatically set to "1", indicate the currently processing interrupt level. Upcoming interrrupts request to the CPU which have lower priority(lower level) will be disabled. When the multiple bits are "1" in the CIL, they indicate the CPU is processing the multiple interrupts.

Each bit of CIL is automatically set to "1", so it has to be cleared 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. See the section "5.3.6 How to program the interrupt process when the interrupt level control is enabled".

Description of bits

- **CILM3-0** (Bit 3 to 0)
  This indicates that a maskable interrupt request of the level corresponding the pit position in which "1" is set is being processed by the processor.

CILM0	Description
0	Processing of Interrupt level 1 not in progress (initial value)
1	Processing of Interrupt level 1 in progress

CILM1	Description
0	Processing of Interrupt level 2 not in progress (initial value)
1	Processing of Interrupt level 2 in progress

CILM2	Description
0	Processing of Interrupt level 3 not in progress (initial value)
1	Processing of Interrupt level 3 in progress

CILM3	Description
0	Processing of Interrupt level 4 not in progress (initial value)
1	Processing of Interrupt level 4 in progress

- **CILN** (Bit 7)
  This indicates that an interrupt request of the highest level or a non-maskable interrupt is being processed by the processor.

CILN	Description
0	Processing of highest level interrupt not in progress (initial value)
1	Processing of highest level interrupt in progress

5.2.12 Interrupt Level Control Register 0 (ILC0)

Address: 0xF034
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC0															
Byte symbol	ILC01								ILC00							
Bit symbol	-	-	ILVLS0H	ILVLS0L	-	-	-	-	-	-	-	-	-	-	-	-
Access type	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

Interrupt level control register 0 (ILC0) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	-	-	-
Bit 13 and 12	ILVLS0H	ILVLS0L	VLS0 interrupt (VLS0INT)
Bit 11 and 10	-	-	-
Bit 9 and 8	-	-	-
Bit 7 and 6	-	-	-
Bit 5 and 4	-	-	-
Bit 3 and 2	-	-	-
Bit 1 and 0	-	-	-

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~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 Interrupt Level Control Register 1 (ILC1)

Address: 0xF036
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC1															
Byte symbol	ILC11								ILC10							
Bit symbol	ILPI7H	ILPI7L	ILPI6H	ILPI6L	ILPI5H	ILPI5L	ILPI4H	ILPI4L	ILPI3H	ILPI3L	ILPI2H	ILPI2L	ILPI1H	ILPI1L	ILPI0H	ILPI0L
Access type	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

Interrupt level control register 1 (ILC1) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	ILPI7H	ILPI7L	Pin interrupt 7 (EXI7INT)
Bit 13 and 12	ILPI6H	ILPI6L	Pin interrupt 6 (EXI6INT)
Bit 11 and 10	ILPI5H	ILPI5L	Pin interrupt 5 (EXI5INT)
Bit 9 and 8	ILPI4H	ILPI4L	Pin interrupt 4 (EXI4INT)
Bit 7 and 6	ILPI3H	ILPI3L	Pin interrupt 3 (EXI3INT)
Bit 5 and 4	ILPI2H	ILPI2L	Pin interrupt 2 (EXI2INT)
Bit 3 and 2	ILPI1H	ILPI1L	Pin interrupt 1 (EXI1INT)
Bit 1 and 0	ILPI0H	ILPI0L	Pin interrupt 0 (EXI0INT)

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~IE7 registers are “0”) or master interrupt enable flag(MIE) is “0”, otherwise, an interrupt may occur with an unexpected interrupt level.

5.2.14 Interrupt Level Control Register 2 (ILC2)

Address: 0xF038
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC2															
Byte symbol	ILC21								ILC20							
Bit symbol	-	-	ILSADH	ILSADL	-	-	ILSIU01H	ILSIU01L	ILSIU00H	ILSIU00L	-	-	ILDMAH	ILDMAL	-	-
Access type	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	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

Interrupt level control register 2 (ILC2) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	-	-	-
Bit 13 and 12	ILSADH	ILSADL	Successive approximation type A/D interrupt (SADINT)
Bit 11 and 10	-	-	-
Bit 9 and 8	ILSIU01H	ILSIU01L	Serial unit 01 interrupt (SIU01INT)
Bit 7 and 6	ILSIU00H	ILSIU00L	Serial unit 00 interrupt (SIU00INT)
Bit 5 and 4	-	-	-
Bit 3 and 2	ILDMAH	ILDMAL	DMAC interrupt (DMACINT)
Bit 1 and 0	-	-	-

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~IE7 registers are “0”) or master interrupt enable flag(MIE) is “0”, otherwise, an interrupt may occur with an unexpected interrupt level.

5.2.15 Interrupt Level Control Register 3 (ILC3)

Address: 0xF03A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC3															
Byte symbol	ILC31								ILC30							
Bit symbol	ILTM1H	ILTM1L	ILTM0H	ILTM0L	ILFTM1H	ILFTM1L	ILFTM0H	ILFTM0L	-	-	ILI2CM0H	ILI2CM0L	-	-	-	-
Access type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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

Interrupt level control register 0 (ILC3) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	ILTM1H	ILTM1L	16-bit timer 1 interrupt (TM1INT)
Bit 13 and 12	ILTM0H	ILTM0L	16-bit timer 0 interrupt (TM0INT)
Bit 11 and 10	ILFTM1H	ILFTM1L	Functional timer 1 interrupt (FTM1INT)
Bit 9 and 8	ILFTM0H	ILFTM0L	Functional timer 0 interrupt (FTM0INT)
Bit 7 and 6	-	-	-
Bit 5 and 4	ILI2CM0H	ILI2CM0L	I2C master 0 interrupt (I2CM0INT)
Bit 3 and 2	-	-	-
Bit 1 and 0	-	-	-

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~IE7 registers are “0”) or master interrupt enable flag(MIE) is “0”, otherwise, an interrupt may occur with an unexpected interrupt level.

5.2.16 Interrupt Level Control Register 4 (ILC4)

Address: 0xF03C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC4															
Byte symbol	ILC41								ILC40							
Bit symbol	ILTM3H	ILTM3L	ILTM2H	ILTM2L	ILFTM3H	ILFTM3L	ILFTM2H	ILFTM2L	-	-	ILSIU11H	ILSIU11L	ILSIU10H	ILSIU10L	ILI2CU0H	ILI2CU0L
Access type	R/W	R/W	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

Interrupt level control register 4 (ILC4) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	ILTM3H	ILTM3L	16-bit timer 3 interrupt (TM3INT)
Bit 13 and 12	ILTM2H	ILTM2L	16-bit timer 2 interrupt (TM2INT)
Bit 11 and 10	ILFTM3H	ILFTM3L	Functional timer 3 interrupt (FTM3INT)
Bit 9 and 8	ILFTM2H	ILFTM2L	Functional timer 2 interrupt (FTM2INT)
Bit 7 and 6	-	-	-
Bit 5 and 4	ILSIU11H	ILSIU11L	Serial unit 11 interrupt (SIU11INT)
Bit 3 and 2	ILSIU10H	ILSIU10L	Serial unit 10 interrupt (SIU10INT)
Bit 1 and 0	ILI2CU0H	ILI2CU0L	I2C master/slave 0 interrupt (I2CU0INT)

[Note]
- Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0~IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.

5.2.17 Interrupt Level Control Register 5 (ILC5)

Address: 0xF03E
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC5															
Byte symbol	ILC51								ILC50							
Bit symbol	ILTM5H	ILTM5L	ILTM4H	ILTM4L	-	-	-	-	-	-	ILCMP0H	ILCMP0L	-	-	-	-
Access type	R/W	R/W	R/W	R/W	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

Interrupt level control register 5 (ILC5) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	ILTM5H	ILTM5L	16-bit timer 5 interrupt (TM5INT)
Bit 13 and 12	ILTM4H	ILTM4L	16-bit timer 4 interrupt (TM4INT)
Bit 11 and 10	-	-	-
Bit 9 and 8	-	-	-
Bit 7 and 6	-	-	-
Bit 5 and 4	ILCMP0H	ILCMP0L	Comparator interrupt (CMP0INT)
Bit 3 and 2	-	-	-
Bit 1 and 0	-	-	-

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~IE7 registers are “0”) or master interrupt enable flag(MIE) is “0”, otherwise, an interrupt may occur with an unexpected interrupt level.

5.2.18 Interrupt Level Control Register 6 (ILC6)

Address: 0xF040
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC6															
Byte symbol	ILC61								ILC60							
Bit symbol	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'
Access type	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

Interrupt level control register 6 (ILC6) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

This register is for feature expansion in futrue products and not used in ML62Q1200A group.

5.2.19 Interrupt Level Control Register 7 (ILC7)

Address: 0xF042
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	ILC7															
Byte symbol	ILC71								ILC70							
Bit symbol	-	-	-	-	ILLTBC2H	ILLTBC2L	ILLTBC1H	ILLTBC1L	-	-	ILLTBC0H	ILLTBC0L	-	-	-	-
Access type	R	R	R	R	R/W	R/W	R/W	R/W	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

Interrupt level control register 7 (ILC7) is a specific function register (SFR) to set the interrupt level for each maskable interrupt source. It is accessible only when the interrupt level control is enabled by the interrupt level control enable register (ILEN).

Description of setting value

Setting value		Description
0	0	Level 1 (interrupt level low) (initial value)
0	1	Level 2
1	0	Level 3
1	1	Level 4 (interrupt level high)

Corresponding block

Bit	Bit symbol name		Corresponding interrupt
Bit 15 and 14	-	-	-
Bit 13 and 12	-	-	-
Bit 11 and 10	ILLTBC2H	ILLTBC2L	Time base counter 2 interrupt (LTBC2INT)
Bit 9 and 8	ILLTBC1H	ILLTBC1L	Time base counter 1 interrupt (LTBC1INT)
Bit 7 and 6	-	-	-
Bit 5 and 4	ILLTBC0H	ILLTBC0L	Time base counter 0 interrupt (LTBC0INT)
Bit 3 and 2	-	-	-
Bit 1 and 0	-	-	-

[Note]
- Enable the interrupt level control function by setting the ILE bit to “1” when the interrupt is disabled(IE0~IE7 registers are “0”) or master interrupt enable flag(MIE) is “0”, otherwise, an interrupt may occur with an unexpected interrupt level.

5.3 Description of Operation

Interrupts of 20 sources except for watchdog timer interrupt (WDTINT) are controlled to be enabled or disabled by the master interrupt enable flag (MIE) of the CPU and interrupt enable registers (IE1 to 7). WDTINT is a non-maskable interrupt.
When the interrupt conditions are satisfied, the CPU calls a branching destination address from the vector table determined for each interrupt source and the interrupt shift cycle starts to branch to the interrupt processing routine.

When the interrupt level control function is disabled and multiple interrupts simultaneously occurred, the CPU handles in order of highest priority defined by the interrupt source no. (smallest interrupt source no. has the highest priority) and holds interrupts with lower priority.
When the interrupt level control function is enabled and multiple interrupts simultaneously occurred, the CPU handles in order of highest priority defined by the interrupt source no. (smallest interrupt source no. has the highest priority) and interrupt level (largest no. has highest priority), and holds interrupts with lower priority.

Table 5-1 lists the interrupt sources.
The interrupt vector address is an address of the interrupt vector area defined in the flash memory. For details, see "nX-U16/100 Core Instruction Manual."

Table 5-1 List of Interrupt Sources (1/2)

Interrupt Source No.	Register assignment			Interrupt vector address	Mask	Internal/ external	Interrupt source	Interrupt source symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
1 (highest priority)	IRQ0[0]	-	-	0x0008	Non maskable	Internal source	WDT interrupt	WDTINT
2	-	-	-	0x000A			-	-
3	IRQ0[6]	IE0[6]	ILC0[13:12]	0x000C	Maskable		VLS interrupt	VLS0INT
4	IRQ0[7]	IE0[7]	ILC0[15:14]	0x000E			-	-
5	IRQ1[0]	IE1[0]	ILC1[1:0]	0x0010	Maskable	External pin	Pin interrupt 0	EXI0INT
6	IRQ1[1]	IE1[1]	ILC1[3:2]	0x0012			Pin interrupt 1	EXI1INT
7	IRQ1[2]	IE1[2]	ILC1[5:4]	0x0014			Pin interrupt 2	EXI2INT
8	IRQ1[3]	IE1[3]	ILC1[7:6]	0x0016			Pin interrupt 3	EXI3INT
9	IRQ1[4]	IE1[4]	ILC1[9:8]	0x0018			Pin interrupt 4	EXI4INT
10	IRQ1[5]	IE1[5]	ILC1[11:10]	0x001A			Pin interrupt 5	EXI5INT
11	IRQ1[6]	IE1[6]	ILC1[13:12]	0x001C			Pin interrupt 6	EXI6INT
12	IRQ1[7]	IE1[7]	ILC1[15:14]	0x001E			Pin interrupt 7	EXI7INT
13	IRQ2[0]	IE2[0]	ILC2[1:0]	0x0020	Maskable	Internal source	-	-
14	IRQ2[1]	IE2[1]	ILC2[3:2]	0x0022			DMAC interrupt	DMACINT
15	IRQ2[2]	IE2[2]	ILC2[5:4]	0x0024			-	-
16	IRQ2[3]	IE2[3]	ILC2[7:6]	0x0026			SIU00 interrupt	SIU00INT
17	IRQ2[4]	IE2[4]	ILC2[9:8]	0x0028			SIU01 interrupt	SIU01INT
18	IRQ2[5]	IE2[5]	ILC2[11:10]	0x002A			-	-
19	IRQ2[6]	IE2[6]	ILC2[13:12]	0x002C			SA-ADC interrupt	SADINT
20	IRQ2[7]	IE2[7]	ILC2[15:14]	0x002E			-	-
21	IRQ3[0]	IE3[0]	ILC3[1:0]	0x0030	Maskable	Internal source	-	-
22	IRQ3[1]	IE3[1]	ILC3[3:2]	0x0032			-	-
23	IRQ3[2]	IE3[2]	ILC3[5:4]	0x0034			I2C master 0 interrupt	I2CM0INT
24	IRQ3[3]	IE3[3]	ILC3[7:6]	0x0036			-	-
25	IRQ3[4]	IE3[4]	ILC3[9:8]	0x0038			F timer 0 interrupt	FTM0INT
26	IRQ3[5]	IE3[5]	ILC3[11:10]	0x003A			F timer 1 interrupt	FTM1INT
27	IRQ3[6]	IE3[6]	ILC3[13:12]	0x003C			Timer 0 interrupt	TM0INT
28	IRQ3[7]	IE3[7]	ILC3[15:14]	0x003E			Timer 1 interrupt	TM1INT
29	IRQ4[0]	IE4[0]	ILC4[1:0]	0x0040	Maskable	Internal source	I2C master/slave 0 interrupt	I2CU0INT
30	IRQ4[1]	IE4[1]	ILC4[3:2]	0x0042			SIU10 interrupt	SIU10INT
31	IRQ4[2]	IE4[2]	ILC4[5:4]	0x0044			SIU11 interrupt	SIU11INT
32	IRQ4[3]	IE4[3]	ILC4[7:6]	0x0046			-	-
33	IRQ4[4]	IE4[4]	ILC4[9:8]	0x0048			F timer 2 interrupt	FTM2INT
34	IRQ4[5]	IE4[5]	ILC4[11:10]	0x004A			F timer 3 interrupt	FTM3INT
35	IRQ4[6]	IE4[6]	ILC4[13:12]	0x004C			Timer 2 interrupt	TM2INT
36	IRQ4[7]	IE4[7]	ILC4[15:14]	0x004E			Timer 3 interrupt	TM3INT

Interrupt Source No.	Register assignment			Interrupt vector address	Mask	Internal/ external	Interrupt Source	Interrupt Source symbol
	IRQ (interrupt Request)	IE (interrupt enable)	ILC (interrupt level)					
37	IRQ5[0]	IE5[0]	ILC5[1:0]	0x0050	Enabled	Internal source	-	-
38	IRQ5[1]	IE5[1]	ILC5[3:2]	0x0052	Enabled		-	-
39	IRQ5[2]	IE5[2]	ILC5[5:4]	0x0054	Enabled		Comparator Interrupt	CMP0INT
40	IRQ5[3]	IE5[3]	ILC5[7:6]	0x0056	Enabled		-	-
41	IRQ5[4]	IE5[4]	ILC5[9:8]	0x0058	Enabled		-	-
42	IRQ5[5]	IE5[5]	ILC5[11:10]	0x005A	Enabled		-	-
43	IRQ5[6]	IE5[6]	ILC5[13:12]	0x005C	Enabled		Timer 4 interrupt	TM4INT
44	IRQ5[7]	IE5[7]	ILC5[15:14]	0x005E	Enabled		Timer 5 interrupt	TM5INT
45	IRQ6[0]	IE6[0]	ILC6[1:0]	0x0060	Enabled	Internal source	-	-
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 source	-	-
54	IRQ7[1]	IE7[1]	ILC7[3:2]	0x0072	Enabled		-	-
55	IRQ7[2]	IE7[2]	ILC7[5:4]	0x0074	Enabled		LTBC0 interrupt	LTBC0INT
56	IRQ7[3]	IE7[3]	ILC7[7:6]	0x0076	Enabled		-	-
57	IRQ7[4]	IE7[4]	ILC7[9:8]	0x0078	Enabled		LTBC1 interrupt	LTBC1INT
58	IRQ7[5]	IE7[5]	ILC7[11:10]	0x007A	Enabled		LTBC2 interrupt	LTBC2INT
59	IRQ7[6]	IE7[6]	ILC7[13:12]	0x007C	Enabled		-	-
60 (low)	IRQ7[7]	IE7[7]	ILC7[15:14]	0x007E	Enabled		-	-

[Note]
- The Watch Dog Timer interrupt (WDTINT) is non maskable. If the non maskable interrupt occurred during an interrupt process the non maskable interrupt have the higest priority and other current interrupts gets hold in spite the multiple interrupt is enabled or disabled.
- For failsafe, define also unused interrupt vectors. If the unused interrupt happened it signifys possibility of that the CPU went out of control. Therefore, it is recommended to make the Watch Dog Timer overflow and reset by program codes execute an infinite loop.

5.3.1 Maskable Interrupt Processing

When the maskable interrupt is generated on the condition of that MIE flag is set to "1", the following processing is performed by hardware and the processing of software shifts to the interrupt destination.

(1) Transfer the program counter (PC) to ELR1
(2) Transfer CSR to ECSR1 (no process when the program memory size is 64Kbyte or smaller)
(3) Transfer PSW to EPSW1
(4) Set the ELEVEL field of PSW to"1"
(5) Set the MIE flag to "0"
(6) Reset CSR to "0" (no process when the program memory size is 64Kbyte or smaller)
(7) Load the value written in the vector table to the program counter (PC)

5.3.2 Non-Maskable Interrupt Processing

When the non-maskkable interrupt is generated, the following processing is performed by hardware and the processing of software shifts to the interrupt destination, regardless of the state of MIE flag.

(1) Transfer the program counter (PC) to ELR2
(2) Transfer CSR to ECSR2 (no process when the program memory size is 64Kbyte or smaller)
(3) Transfer PSW to EPSW2
(4) Set the ELEVEL field of PSW to "2"
(5) Reset CSR to "0" (no process when the program memory size is 64Kbyte or smaller)
(6) Load the value written in the vector table to the program counter (PC)

5.3.3 Software Interrupt Processing

The software interrupt is generated as required in an application program. When the SWI instruction is performed in the program, the software interrupt is generated, the following processing is performed by hardware, and the processing of software shifts to the interrupt destination. The vector table is specified by the SWI instruction.

(1) Transfer the program counter (PC) to ELR1
(2) Transfer CSR to ECSR1 (no process when the program memory size is 64Kbyte or smaller)
(3) Transfer PSW to EPSW1
(4) Set the ELEVEL field of PSW to "1"
(5) Set the MIE flag to "0"
(6) Reset CSR to "0" (no process when the program memory size is 64Kbyte or smaller)
(7) Load the value written in the vector table to the program counter (PC)

For the MIE flag, PC (program counter), CSR, PSW, and ELEVEL, see the separate "nX-U16/100 Core Instruction Manual."

5.3.4 Notes on Interrupt Routine (when the Interrupt Level Control is disabled)

When "0" is written to the interrupt level control enable register (ILEN), the interrupt level control is disabled.
There are some different notes in following states and conditions.

- When the maskable interrupt is being executed(State A) and when the non-maskable interrupt is being executed (State B)
- When the sub routine is called in the interrupt routine or when the sub routine is not called in the interrupt routine
- When the multiple interrupt is disabled or enabled

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

· 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 return the contents of the EPSW register to PSW.

A-1-2: When multiple interrupts are enabled

· 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.

Example of description: State A-1-1

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

Example of description: State A-1-2

```
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
```

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

A-2-1: When multiple interrupts are disabled

- 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.

A-2-2: When multiple interrupts are enabled

- 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.

Example of description:
State A-2-1

```
Intrpt_A-2-1;          ;Start
 PUSH LR               ;Save ELR, EPSW, 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
```

Example of description:
State A-2-2

```
Intrpt_A-2-2;          ;Start
 PUSH ELR, EPSW, LR    ;Save ELR, EPSW, 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 RC from LR
                ;End of subroutine
```

State B: Non-maskable interrupt is being processed

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

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

Example of description:
State B-1

```
Intrpt_B-1;                ;B-1 state
  PUSH ELR, EPSW           ;Save ELR, 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
```

Example of description:
State B-2

```
Intrpt_B-2;                ;Start
  PUSH ELR, EPSW, LR       ;Save ELR, EPSW, 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
```

5.3.5 Flow chart of interrupt processing when the Interrupt Level Control is enabled

Figure 5-1 shows the software processing in the case of interrupt level control is enabled, multiple interrupt is disabled or enabled. When the multiple interrupt is enabled, save ELR1 and EPSW1 to the stack(RAM) so that they are not overwritten by the multiple interrupt. The EI and DI instructions determines whether the execution of overlapped interrupt with a higher-level maskable interrupt request is enable while the "target interrupt processes"(shown in the flow chart) is being executed. If a non-maskable interrupt is occurred as the overlapped interrupt, the non-maskable interrupt will be executed regardless the condition of multiple interrupt and the execution of EI instruction.

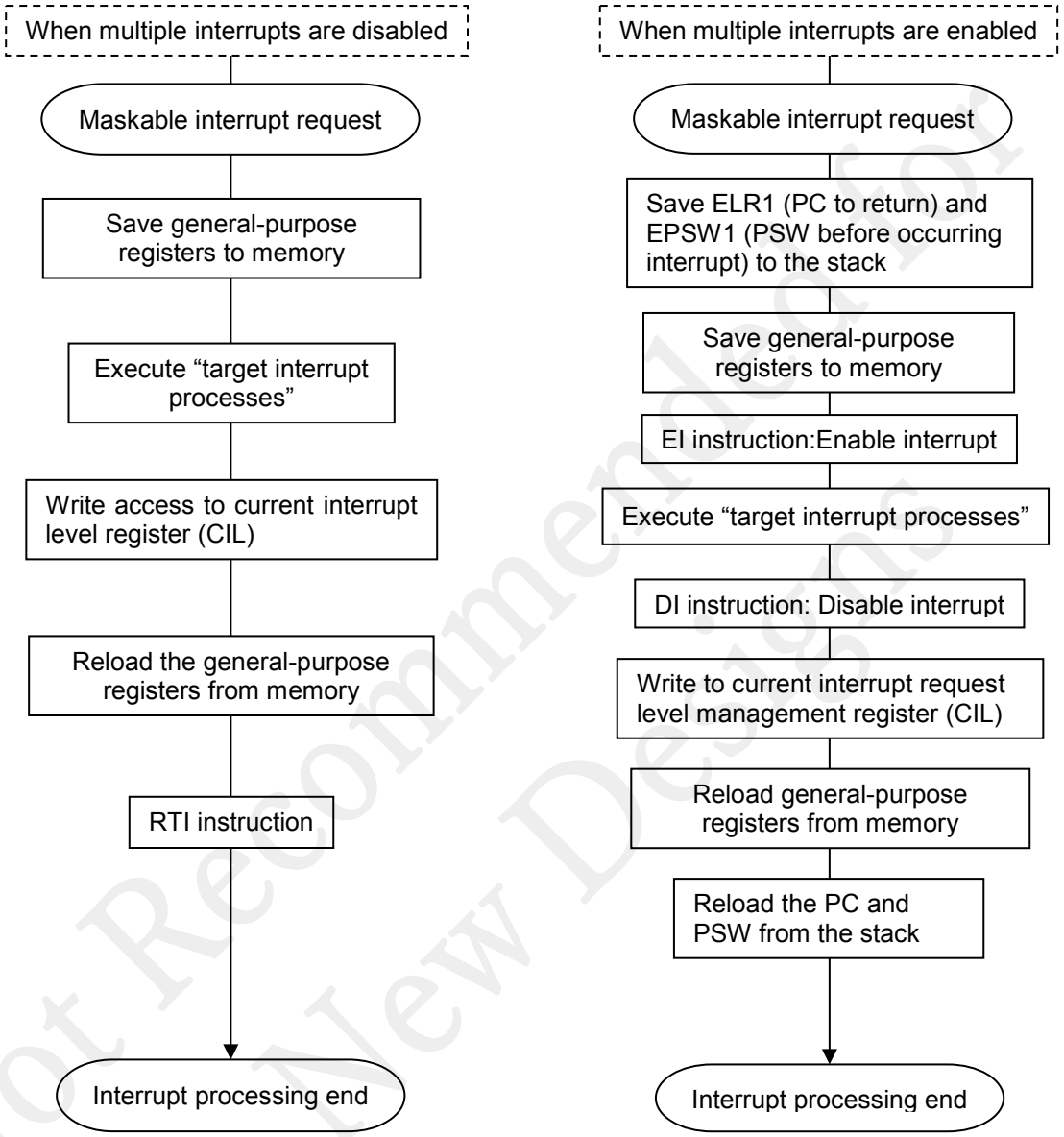


Figure 5-1 Maskable Interrupt Processing Flow

[Note]
- For considering the processing of non-maskable interrupt, refer to the flow charge in the case of the multiple interrupts are enabled. ELR2 and EPSW2 will be saved to the stack in that case.
- When programming by the C language, the program code for the save/reload are generated by the Compiler (Push/Pop instruction). Other codes for enabling/disabling the interrupt by EI/DI instruction and writing to the current interrupt request level management register(CIL).

5.3.6 How to program the interrupt process when the interrupt level control is enabled

When the interrupt level control enable register (ILEN) is set to enable interrupt level control, the interrupt function is described as follows. For a detailed description method of interrupt processing and notes, see "CCU8 Programming Guide."

5.3.6.1 When programming interrupts defined as it disables other multiple interrupts

When programming an interrupt function that does not allow other multiple interrupts, define 1 in the category field of the INTERRUPT pragma and SWI pragma. When the built-in function__EI is called in the interrupt function, the CCU8 outputs an error message.
After completion of target interrupt processes, it is necessary to write to the CIL register and clear the highest current interrupt request level (CILN bit). Otherwise, lower level interrupts will not be acceptable.

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*/
}
```

By programming like the example, intr_fn_0A is handled as an interrupt that does not allow other multiple interrupts. CCU8 generates the following assembly code.

Example of assembly codes generated by the CCU8

```
_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 that may be used in the interrupt processing(only ER0 in this example) is saved to the stack. "RTI" instruction is used to return from the interrupt function.

Following example shows the case that calls an another function 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 an another function is called from an interrupt function, the output program code is redundant and the processing time of the interrupt gets longer, comparing to the case the function is not called. They are because the CCU8 does not know what registers the function func ( ) would use in actual and it saves all registers that has possibility of being changed to the stack.

[Note]
Do not enable any interrupt in a function called from an interrupt function defined as it does not allow the multiple interrupts. Otherwise, the program may run out of control when the multiple interrupts occur.

5.3.6.2 When programming interrupts defined as it enables other multiple interrupts

When programming an interrupt function that allows other multiple interrupts, define 2 in the category field of the INTERRUPT pragma and SWI pragma. Built-in function_EI can be called in the interrupt function defined as it allows the 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*/
}
```

When described as in the example, intr_fn_20 ( ) is handled as an interrupt processing function that allows other multiple interrupts. CCU8 generates the following assembly code.

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 the interrupt function, ELR and EPSW are saved to the stack so that they should not destroyed when the multiple interrupts occurred. This is the difference from the case using the interrupt function defined as disabling other multiple interrupts. Also, instruction "POP PSW, PC" is used instead of "RTI." To return from the interrupt function.

5.3.7 Interrupt Disable State

There are two operating states any interrupt is not accepted even if the interrupt conditions are satisfied. It's called an interrupt disabled state.

State 1. Between the interrupt transfer cycle and the instruction at the beginning of the interrupt routine
When the interrupt conditions are satisfied in that time period, the interrupt will occur immediately after the execution of the instruction at the beginning of the interrupt routine.

State 2. Between the DSR prefix code and the next instruction
When the interrupt conditions are satisfied in that time period, the interrupt will occur immediately after executing the DSR prefix code and the next instruction.

For details about the DSR prefix instruction, see "nX-U16/10 Core Instruction Manual."

Chapter 6 Clock Generation Circuit

6. Clock Generation Circuit

6.1 General Description

ML62Q1000 series has the clock generation circuit that generates following clocks to provide them to the CPU and the peripherals.
See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

- Low-speed clock (LSCLK) : Low speed clock for peripherals (approx. 32.768kHz)
- High-speed clock master (HSCLK) : High speed clock for peripherals (max. 32MHz)
- System clock (SYSTEMCLK) : CPU operating clock (approx. 32.768kHz or max. 24MHz)
  The maximum frequency depends on the CPU operation mode
- High-speed output clock (OUTHSCLK) : Low speed output from a general port (approx. 32.768kHz)
- Low-speed output clock (OUTLSCLK) : High speed output from an general port (max. 12MHz)
- WDT (watchdog timer) clock (WDTCLK) : Clock for watch dog timer (approx. 1.024kHz)

For the output pins of OUTHSCLK and OUTLSCLK, see Chapter 17 "GPIO."

6.1.1 Features

- Low-speed clock generation circuit
  - Low-speed RC oscillation circuit (approx. 32.768 kHz ±3% @-40 to +105°C)
  - Adjustable to ±1% @-40 to +85°C (VDD=1.8V or higher) by using the RC oscillation adjustment sample software

- High-speed clock generation circuit
  - PLL oscillation mode (selectable among 16 MHz/24 MHz/32 MHz by the code option)
  - High-speed clock wake-up time is selectable

- WDT (watchdog timer) clock generation circuit
  - RC1K clock (RC oscillation 1KHz clock for WDT), approx. 1.024kHz -50%~+100%
    The RC1K clock or the 1.024kHz devided from the low-speed RC oscillation clock (32.768 kHz) is selectable by the code option.

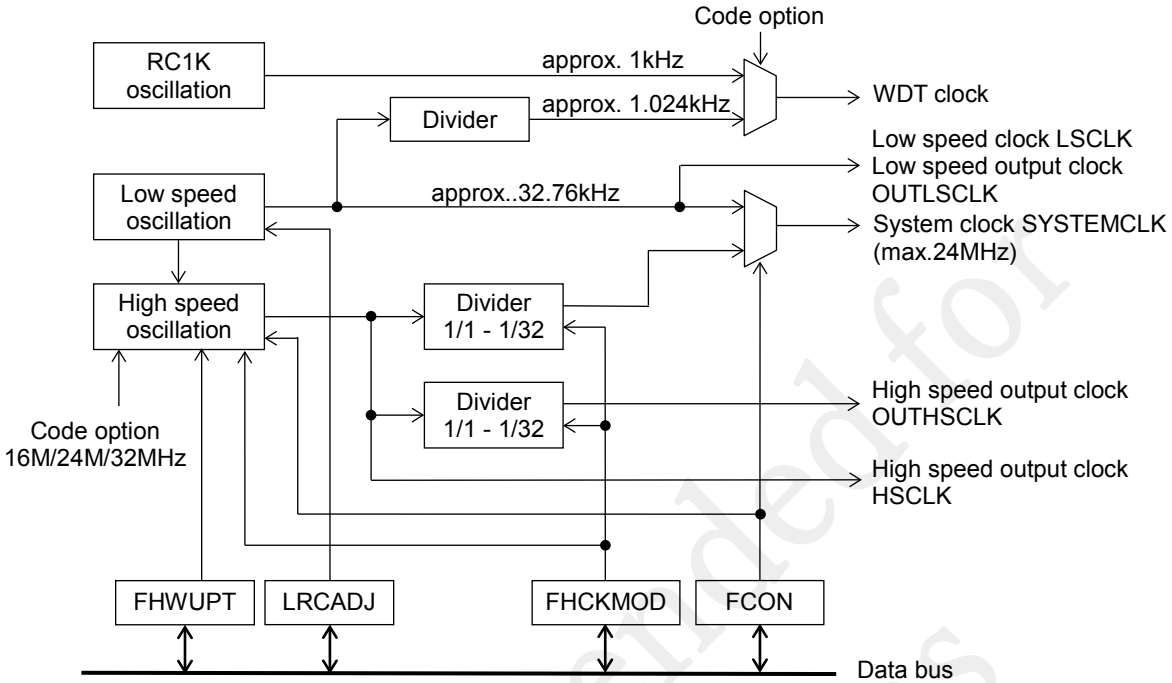
Table 6-1 shows relation of PLL oscillation mode and maximum operating frequency. The PLL oscillation mode and the CPU mode is selectable by the code option. See Chapter 26 “Code Option” for more details.

Table 6-1 PLL oscillation mode and Maximum operating frequency

PLL oscillation mode	Maximum operating frequency		
	Peripheral	CPU (Wait mode)	CPU (No wait mode)
32MHz mode	32MHz	16MHz	8MHz
24MHz mode	24MHz	24MHz	12MHz
16MHz mode	16MHz	16MHz	8MHz

6.1.2 Configuration

Figure 6-1 shows the configuration of the clock generation circuit.
Table 6-2 shows the list of operation clocks for each function.



FHCKMOD : High-speed clock mode register
FCON : Frequency control register
FHWUPT : High-speed clock wake-up time setting register
LRCADJ : Low-speed RC oscillation frequency adjustment register

Figure 6-1 Configuration of Clock Generation Circuit

[Note]
After power-on or system reset, the low-speed RC oscillation clock (approx. 32.768 kHz) is initially selected and supplied to the system clock (SYSTEMCLK).

Table 6-2 Operating clock list in each function

Function	System clock SYSTEMCLK	Low-speed clock LSCLK	High speed clock HSCLK	WDT clock
CPU	○	-	-	-
RAM	○	-	-	-
Watchdog timer	○	-	-	○
External interrupt control *1	○	○	○	-
Time base counter	○	○	-	-
16-bit timer	○	○	○	-
16-bit functional timer	○	○	○	-
Serial communication unit	○	○	○	-
I^2C bus unit	○	○	○	-
I^2C bus master	○	-	○	-
Buzzer	○	○	-	-
A/D converter	○	○	○	-
D/A converter	○	-	-	-
Analog comparator *1	○	○	○	-
Voltage Level Supervisor(VLS) *1	○	○	○	-

○: The clock is supplied -: The clock is Not supplied
*1: The clock is supplied for sampling

6.1.3 List of Pins

The output pins of the high-speed/low-speed clocks are assigned to the second to octic functions of GPIO.
For details of pin assignment and the second to octic settings of GPIO, see Chapter 17 "GPIO."

Pin name	I/O	Function
OUTLSCLK	O	Low-speed clock output
OUTHSCLK	O	High-speed clock output

6.2 Description of Registers

6.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF002	High-speed clock mode register	FHCKMODL	FHCKMOD	R/W	8/16	0x00
0xF003		FHCKMODH		R/W	8	0x44
0xF004	Reserved register	-	-	R	8	0x00
0xF005	Reserved register	-	-	R	8	0x00
0xF006	Clock control register	FCON	-	R/W	8	0x00
0xF007	Reserved register	-	-	R	8	0x00
0xF008	High-speed clock wake-up time setting register	FHWUPT	-	R/W	8	0x00
0xF009	Reserved register	-	-	R	8	0x00
0xF080	Lo-speed RC oscillation frequency adjustment register	LRCADJ	-	R/W	8	0x00

[Note]
For the registers with word symbol, word access is possible. For word access, specify an even address.

6.2.2 High-Speed Clock Mode Register (FHCKMOD)

Address: 0xF002
Access: R/W
Access size: 8/16 bits
Initial value: 0x4400

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FHCKMOD															
Byte symbol	FHCKMODH								FHCKMODL							
Bit symbol	-	OUTC2	OUTC1	OUTC0	-	SYSC2	SYSC1	SYSC0	-	-	-	-	-	-	-	HOSCM0
Access type	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W
Initial value	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0

FHCKMOD is a specific function register (SFR) to select the oscillation mode and frequency of the high-speed clock (HSCLK).

Description of bits

- **HOSCM0** (Bit 0)
  HOSCM0 bit is for selecting the oscillation mode and frequency of the high-speed clock.

HOSCM0	Description
0	PLL oscillation mode (initial value)
1	Do not use (PLL oscillation mode)

- **SYSC2 to SYSC0** (Bit 10 to 8)
  SYSC2 to SYSC0 is for selecting the frequency of the high-speed clock used for the system clock (SYSTEMCLK). At the system reset, 1/16 HSCLK is selected.
  The frequency of system clock is automatically controlled by the hardware, so that it does not exceed the maximum frequency of CPU operating frequency. See the table 6-1 about the maximum frequency.

When the PLL oscillation mode is "32MHz PLL" and CPU operation mode is "Wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	1/2 HSCLK	(=16MHz)
0	0	1	1/2 HSCLK	(=16MHz)
0	1	0	1/4 HSCLK	(=8MHz)
0	1	1	1/8 HSCLK	(=4MHz)
1	0	0	1/16 HSCLK (initial value)	(=2MHz)
1	0	1	1/32 HSCLK	(=1MHz)
1	1	0	Do not use (1/32 HSCLK)	(=1MHz)
1	1	1	Do not use (1/32 HSCLK)	(=1MHz)

When the PLL oscillation mode is "32MHz PLL" and CPU operation mode is "No wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	1/4 HSCLK	(=8MHz)
0	0	1	1/4 HSCLK	(=8MHz)
0	1	0	1/4 HSCLK	(=8MHz)
0	1	1	1/8 HSCLK	(=4MHz)
1	0	0	1/16 HSCLK (initial value)	(=2MHz)
1	0	1	1/32 HSCLK	(=1MHz)
1	1	0	Do not use (1/32 HSCLK)	(=1MHz)
1	1	1	Do not use (1/32 HSCLK)	(=1MHz)

When the PLL oscillation mode is "24MHz PLL" and CPU operation mode is "Wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	HSCLK	(=24MHz)
0	0	1	1/2 HSCLK	(=12MHz)
0	1	0	1/4 HSCLK	(=6MHz)
0	1	1	1/8 HSCLK	(=3MHz)
1	0	0	1/16 HSCLK (initial value)	(=1.5MHz)
1	0	1	1/32 HSCLK	(=0.75MHz)
1	1	0	Do not use (1/32 HSCLK)	(=0.75MHz)
1	1	1	Do not use (1/32 HSCLK)	(=0.75MHz)

When the PLL oscillation mode is "24MHz PLL" and CPU operation mode is "No wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	1/2 HSCLK	(=12MHz)
0	0	1	1/2 HSCLK	(=12MHz)
0	1	0	1/4 HSCLK	(=6MHz)
0	1	1	1/8 HSCLK	(=3MHz)
1	0	0	1/16 HSCLK (initial value)	(=1.5MHz)
1	0	1	1/32 HSCLK	(=0.75MHz)
1	1	0	Do not use (1/32 HSCLK)	(=0.75MHz)
1	1	1	Do not use (1/32 HSCLK)	(=0.75MHz)

When the PLL oscillation mode is "16MHz PLL" and CPU operation mode is "Wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	HSCLK	(=16MHz)
0	0	1	1/2 HSCLK	(=8MHz)
0	1	0	1/4 HSCLK	(=4MHz)
0	1	1	1/8 HSCLK	(=2MHz)
1	0	0	1/16 HSCLK (initial value)	(=1MHz)
1	0	1	1/32 HSCLK	(=0.5MHz)
1	1	0	Do not use (1/32 HSCLK)	(=0.5MHz)
1	1	1	Do not use (1/32 HSCLK)	(=0.5MHz)

When the PLL oscillation mode is "16MHz PLL" and CPU operation mode is "No wait mode"

SYSC2	SYSC1	SYSC0	Description	
0	0	0	1/2 HSCLK	(=8MHz)
0	0	1	1/2 HSCLK	(=8MHz)
0	1	0	1/4 HSCLK	(=4MHz)
0	1	1	1/8 HSCLK	(=2MHz)
1	0	0	1/16 HSCLK (initial value)	(=1MHz)
1	0	1	1/32 HSCLK	(=0.5MHz)
1	1	0	Do not use (1/32 HSCLK)	(=0.5MHz)
1	1	1	Do not use (1/32 HSCLK)	(=0.5MHz)

- **OUTC2 to OUTC0** (Bit 14 to 12)
  OUTC2 to OUTC0 is for selecting the frequency of the high-speed output clock (OUTHSCLK) output from the general port. At the system reset, 1/16 HSCLK is selected.

OUTC2	OUTC1	OUTC0	Description	CPU mode: Wait mode or No wait mode		
				PLL mode: 16MHz	PLL mode: 24MHz	PLL mode: 32MHz
0	0	0	Do not use	-	-	-
0	0	1	1/2 HSCLK	8MHz	12MHz	16MHz
0	1	0	1/4 HSCLK	4MHz	6MHz	8MHz
0	1	1	1/8 HSCLK	2MHz	3MHz	4MHz
1	0	0	1/16 HSCLK (initial value)	1MHz	1.5MHz	2MHz
1	0	1	1/32 HSCLK	0.5MHz	0.75MHz	1MHz
1	1	0	Do not use	-	-	-
1	1	1	Do not use	-	-	-

[Note]
- When the voltage of V_{DD} is $1.6V \leqq V_{DD} < 1.8V$, set the system clock to 4 MHz or lower. If it exceeds 4 MHz, the operation is not guaranteed.
- For output of the high-speed clock (OUTHSCLK), the output clock frequency is limited according to the voltage of V_{DD}.
  $1.6V \leqq V_{DD} < 1.8V$: Select 4 MHz or lower
  $1.8V \leqq V_{DD} \leqq 5.5V$: Select 12 MHz or lower

6.2.3 Clock Control Register (FCON)

Address: 0xF006
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FCON							
Bit symbol	-	-	-	-	-	-	-	-	LPLL	-	-	-	-	-	ENOSC	SELSCLK
Access type	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

FCON is a specific function register (SFR) to select control of the clock generation circuit and the system clock.

Description of bits

- **SELSCLK** (Bit 0)
  SELSCLK sets the clock supplied to the system clock. The low-speed clock (LSCLK) or HSCLK (1/n HSCLK: n = 1, 2, 4, 8, 16, 32) selected in the high-speed clock frequency selection bit (SYSC2, 1, 0) of FHCKMOD is selectable for the system clock.
  When the high-speed generation circuit is stopped (ENOSC bit = “0”), the SELSCLK bit is fixed to "0" and the low-speed clock (LSCLK) is selected for the system clock.

SELSCLK	Description
0	Low-speed clock (LSCLK) is selected for the system clock (SYSTEMCLK)
1	High-speed clock (HSCLK ~ 1/32 HSCLK) is selected for the system clock (SYSTEMCLK)

- **ENOSC** (Bit 1)
  ENOSC sets whether to enable or disable the oscillation of the high-speed clock oscillation circuit.

ENOSC	Description
0	Stop the high-speed oscillation (initial value)
1	Enable/Start the high-speed oscillation

- **LPLL** (Bit 7)
  LPLL indicates that the frequency of the PLL oscillation is withing the specified error. The LPLL has the read-only attribute.

LPLL	Description
0	Output frequency of PLL oscillation circuit not stabilized or PLL oscillation
1	Output frequency of PLL oscillation circuit stabilized

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

Address: 0xF008
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FHWUPT							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FHUT0
Access type	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

FHWUPT is a specific function register (SFR) to select the wake-up time of the high-speed clock.
FHWUPT is writable when the high-speed oscillation is stopped.
Bit 1 to Bit 15 are reserved bits. Write “0” to the Bit1 to Bit15.

Description of bits

- **FHUT0** (Bit 0)
  FHUT0 is for selecting the wake-up time of the high-speed clock.
  The clock is supplied to the CPU after it’s stabilized or before it’s stabilized. In the case the clock is supplied before it’s stabilized, the clock is supplied to the CPU approx. 30µs after enabling/starting the high speed oscillation and the clock frequency is guradually getting higher and reaches to the target frequency in approx. 2ms. The frequency in the approx.2ms can not be guaranteed as the specification, but it is useable for the system clock.

  The FHUT0 is writeable when the high-speed oscillation is stopped.
  The wake-up time from the STOP mode and STOP-D mode is different. See the table 4-4 “Wake-up Times from STOP/STOP-D Mode” for details about it.

FHUT0	Description
0	The clock is supplied after it’s stabilized: approx. 2.5 ms (initial value)
1	The clock is supplied before it’s stabilized: approx. 30 µs

[Note]
Write always "0" to FHWUPT [7:1]. Otherwise, the operation is not guaranteed.

6.2.5 Low-Speed RC Oscillation Frequency Adjustment Register (LRCADJ)

Address: 0xF080
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								LRCADJ							
Bit symbol	-	-	-	-	-	-	-	-	-	-	LRCADJ5	LRCADJ4	LRCADJ3	LRCADJ2	LRCADJ1	LRCADJ0
Access type	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

LRCADJ is a specific function register (SFR) to adjust the frequency of the low-speed RC oscillation clock.
Use the RC oscillation adjustment sample software provided by LAPIS.
As the low-speed RC oscillation circuit has temperature variation, the software performs the frequency adjustment according to the temperature by using the frequency adjustment function in the low-speed RC oscillation circuit. The software calculates a trimming value using following three parameters and primary approximate equation in the area of low temperature - ordinary temperature and ordinary temperature - high temperature, and then sets it to the low-speed RC oscillation frequency adjustment register.
[parameter]
1.Temperature sensor A/D conversion value of the current temperature, for the low-speed RC oscillation frequency Adjustment
2.Temperature sensor A/D conversion value of the ordinary temperature (25 oC), for the low-speed RC oscillation frequency
Adjustment
3.Coefficient

[Note]
Use the RC oscillation adjustment sample software provided by LAPIS. Otherwise, the operation is not guaranteed.

6.3 Description of Operation

6.3.1 Low-Speed Clock

The low-speed generation circuit has the low-speed RC oscillation circuit (approx. 32.768 kHz).

6.3.1.1 Low-Speed RC Oscillation Circuit Configuration

The low-speed RC oscillation circuit is initially selected at the power up and the CPU starts operating after 512 counts of the low-speed RC oscillation clock.
When the STOP/STOP-D mode is released, the CPU starts operating after 8 counts of the low-speed RC clock.
The frequency of the low-speed RC oscillation circuit can be adjusted with the low-speed RC oscillation frequency adjustment register (LRCADJ). Use the RC oscillation adjustment sample software provided by LAPIS, which measures the temperature of the LSI by using the successive approximation type A/D converter and set an adjustment value to the LRCADJ.
Figure 6-2 shows the configuration of the low-speed RC oscillation circuit.
Figure 6-3 shows the operation waveforms at the start of the low-speed RC oscillation circuit and in the STOP/STOP-D mode.

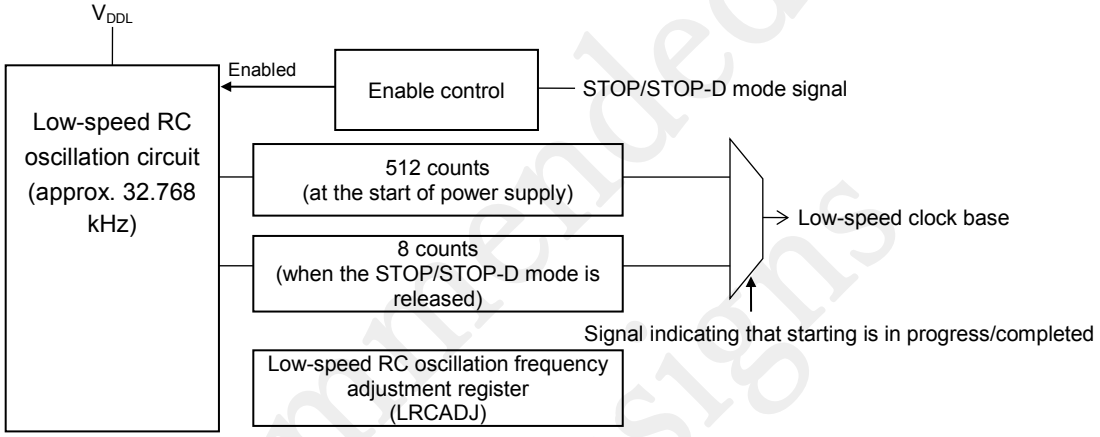


Figure 6-2 Low-Speed RC Oscillation Circuit Configuration

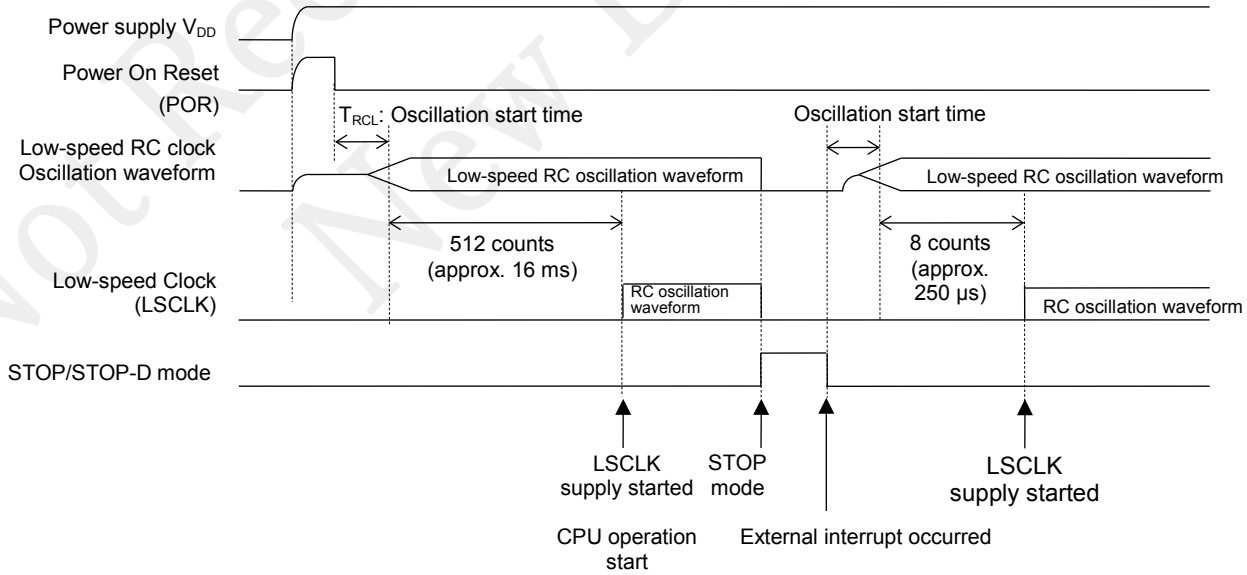


Figure 6-3 Operation Waveforms of Low-Speed RC Oscillation Circuit

6.3.2 High-Speed Clock

The high-speed clock generation circuit has the PLL oscillation circuit installed.
For the oscillation frequency of PLL, 32 MHz, 24 MHz, or 16 MHz can be selected with code option.
The high-speed output clock(OUTHSCLK) can be outputte from a gneral port.

6.3.2.1 PLL Oscillation Circuit Configuration

The PLL oscillation circuit generates the PLL oscillation clock based on the low-speed clock (LSCLK).
After high-speed oscillation is enabled (set ENOSC of FCON to "1") and counts for stabilizing the PLL oscillation clock, the HSCLK (high-speed clock) is output.
The PLL oscillation circuit stops oscillation when the mode is entered to the HALT-H/STOP/STOP-D mode by software.
The clock is supplied to the CPU after it's stabilized or before it's stabilized. In the case the clock is supplied before it's stabilized, the clock is supplied to the CPU approx. 30µs after enabling/starting the high speed oscillation and the clock frequency is guradually getting higher and reaches to the target frequency in approx. 2ms. The frequency in the approx. 2ms can not be specified, but it is useable for the system clock.

The wake-up time from the STOP/STOP-D mode requires the time for at least 8 counts of the low-speed oscillation clock.
For details, see Chapter 4 "Power Management."
Figure 6-4 shows the configuration of the PLL oscillation circuit.
Figure 6-5 shows the operation waveforms at the start of the PLL oscillation circuit and in the STOP/STOP-D mode.

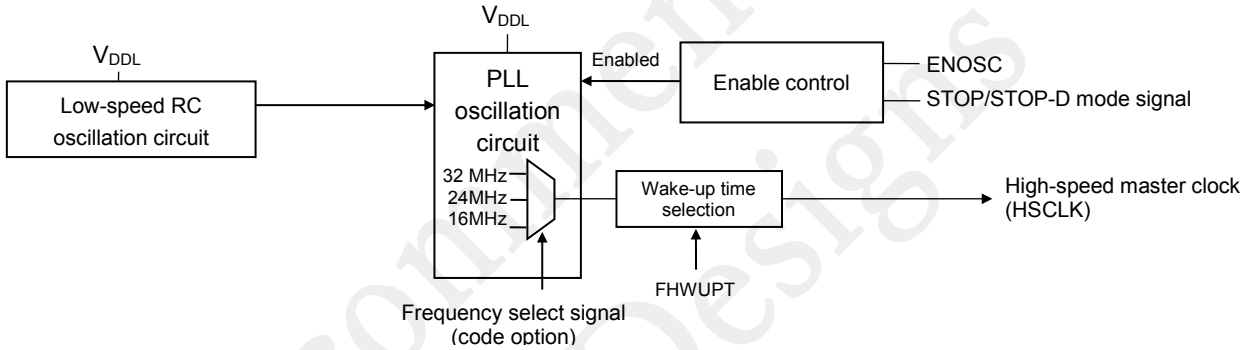


Figure 6-4 PLL Oscillation Circuit Configuration

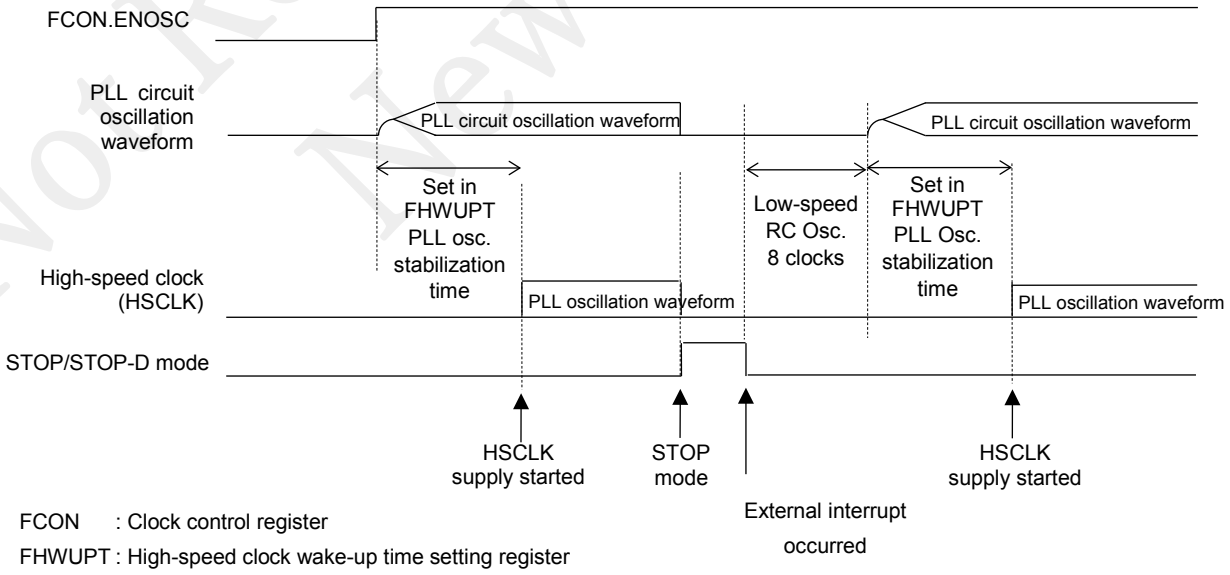


Figure 6-5 Operation Waveforms of PLL Oscillation Circuit

6.3.3 WDT Clock

The 1.024kHz clock devided from the low-speed RC oscillation clock (32.768 kHz) or the 1kHz WDT dedicated RC oscillation clock is selectable by the code option. If accuracy of the frequency is required, select the low-speed RC oscillation circuit.
The WDT operation clock stops the oscillating in the STOP/STOP-D mode.
When the STOP/STOP-D mode is released, the WDT clock will be supplied after a stabilization time (one count of the RC1K clock).
See Chapter 26 "Code Option" for more details about the code option.
Figure 6-6 shows the configuration of the WDT RC oscillation circuit.
Figure 6-7 shows the operation waveforms at the start of the WDT RC oscillation circuit and in the STOP/STOP-D mode.

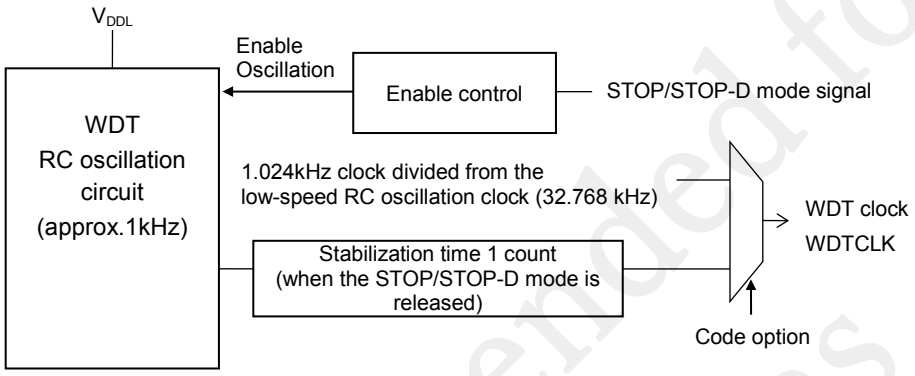


Figure 6-6 WDT dedicated RC Oscillation Circuit Configuration

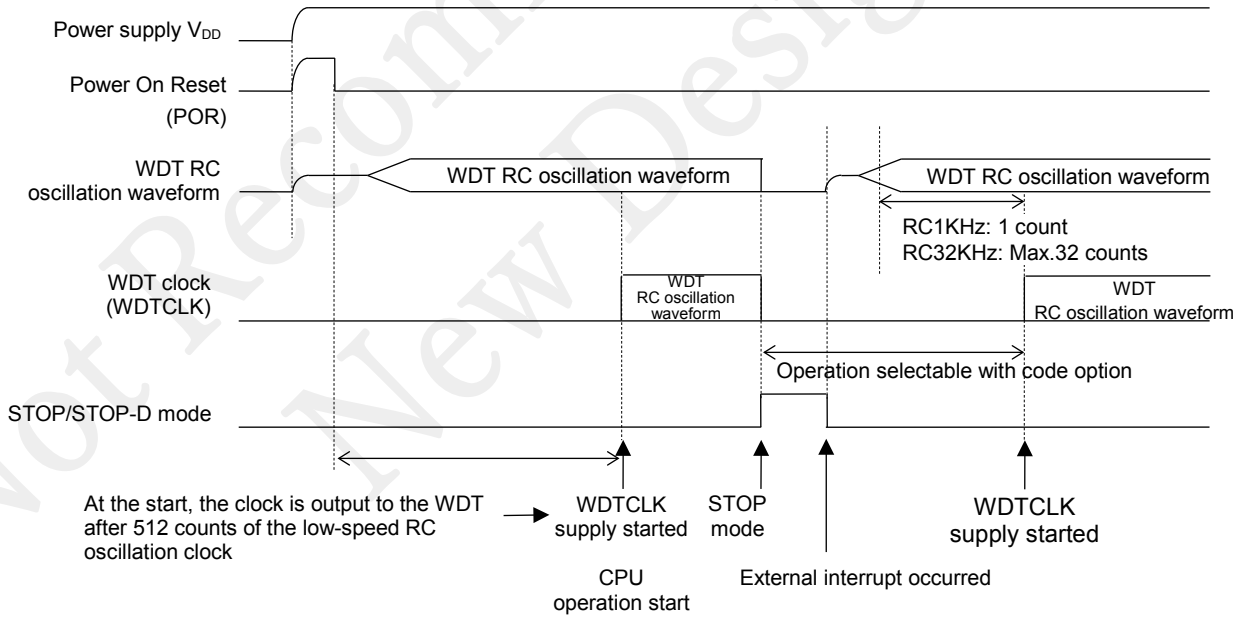


Figure 6-7 Operation Waveforms of WDT dedicated RC Oscillation Circuit

6.3.4 Switching of System Clock

Figure 6-8 shows the flow chart of the system clock switching (LSCLK -> HSCLK).

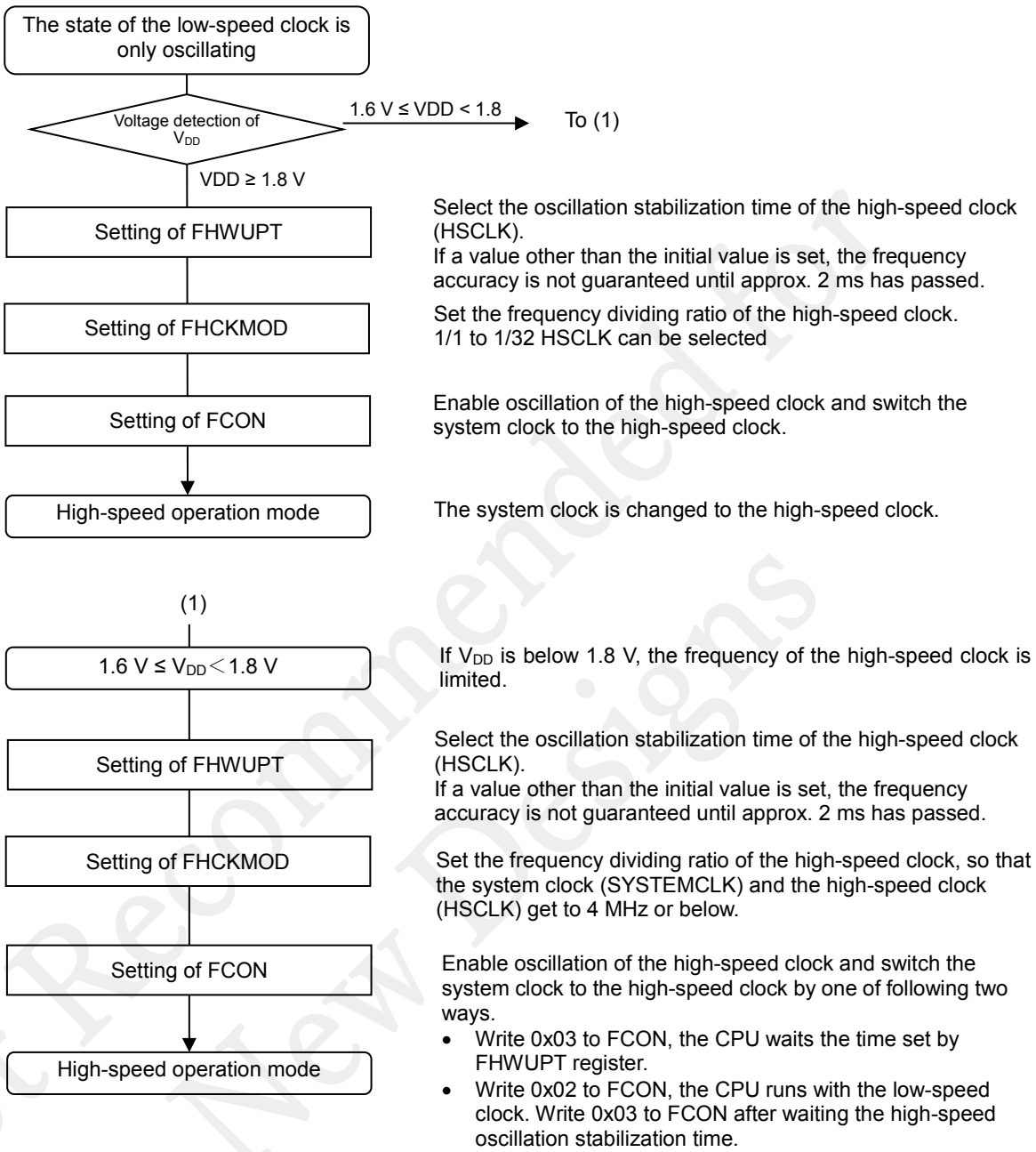


Figure 6-8 Flow Chart of System Clock Switching (LSCLK -> HSCLK)

[Note]
When the voltage of V_{DD} is 1.6 V ≤ VDD < 1.8 V, set the system clock (SYSTEMCLK) and the high-speed clock (HSCLK) to 4 MHz or below. If it exceeds 4 MHz, the operation is not guaranteed.

Figure 6-9 shows the flow chart of the system clock switching (HSCLK -> LSCLK).

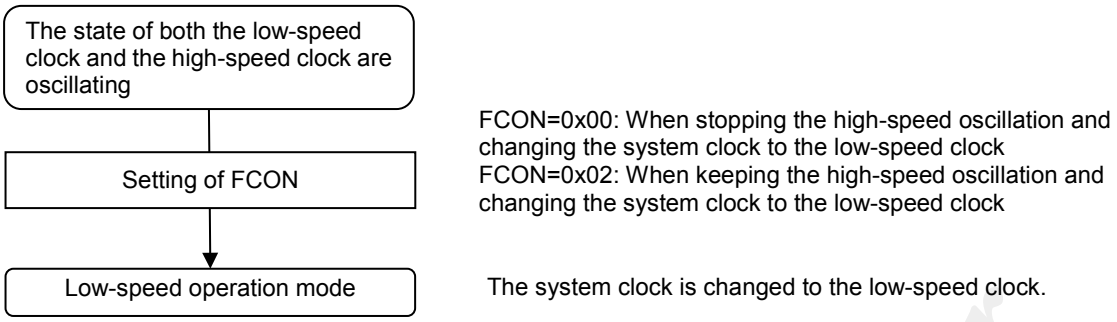


Figure 6-9 Flow Chart of System Clock Switching (HSCLK -> LSCLK)

[Note]
While the system clock is operating at the low speed, if interrupts of the peripheral circuits are enabled for high-speed clock operation, the interrupt processing of the CPU may not be in time. Consider the timing of the interrupt cycles and the operating frequency of the CPU.

Chapter 7 Low Speed Time Base Counter

7. Low Speed Time Base Counter

7.1 General Description

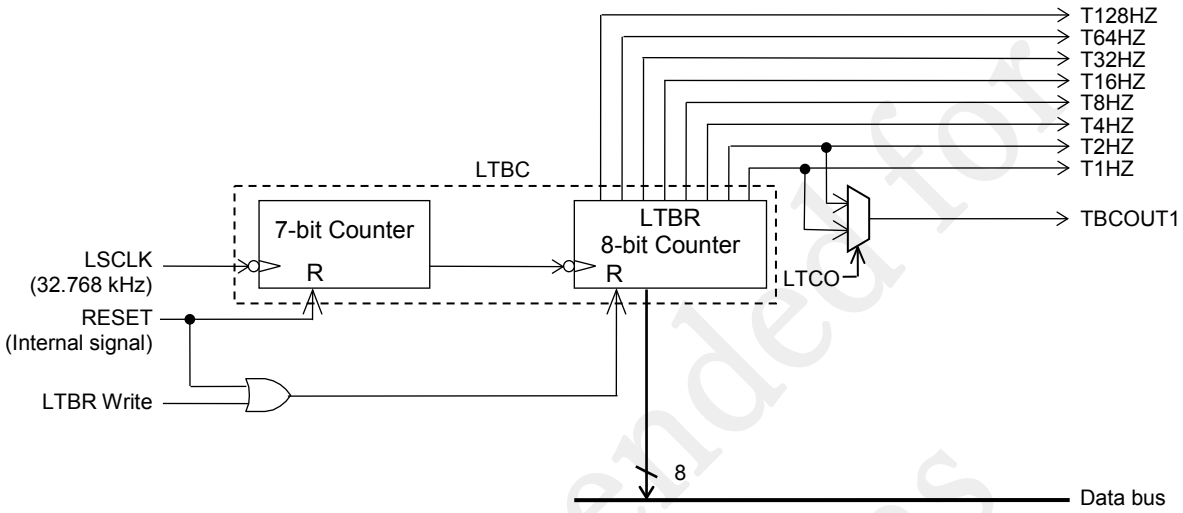
ML62Q1000 series has the low speed time base counter (LTBC) generates base clocks and periodic interrupts for peripheral circuits. For input clocks, see Chapter 6 "Clock Generation Circuit". For the interrupt enable/request flags, etc. described in this chapter, see Chapter 5 "Interrupts."

7.1.1 Features

- Generates 128Hz to 1Hz pulse signals by dividing the low-speed clock (LSCLK) frequency.
- Three interrupts are selectable among eight periodical interrupt request (T28 Hz, 64Hz, 32Hz, 16Hz, 8Hz, 4Hz, 2Hz and 1Hz).
- Clocks with adjusted frequency (1Hz/2Hz) can be output from the pin (TBCOUT1).

7.1.2 Configuration

Figure 7-1 shows the configuration of the low speed time base counter.



LTBR : Low speed time base counter register
LTBINT : Low speed time base counter interrupt selection register
LTBCCON : Low speed time base counter control register

Figure 7-1 Configuration of Low Speed Time Base Counter

7.1.3 List of Pins

The output pins of the low speed time base counter are assigned to the second to octic functions of GPIO.
For details of pin assignment and the second to octic settings of GPIO, see Chapter 17 "GPIO."

Signal name	I/O	Function
TBCOUT1	O	Low speed time base clock (T1HZ/T2HZ) output

7.2 Description of Registers

7.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF060	Low Speed time base counter register	LTBR	-	R/W	8	0x00
0xF061	Reserved register	-	-	R	8	0x00
0xF062	Low Speed time base counter control register	LTBCCON	-	R/W	8	0x01
0xF063	Reserved register	-	-	R	8	0x00
0xF064	Reserved register	-	-	R	8	0x00
0xF065	Reserved register	-	-	R	8	0x00
0xF066	Reserved register	-	-	R	8	0x00
0xF067	Reserved register	-	-	R	8	0x00
0xF068	Low Speed time base counter interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x00
0xF069		LTBINTH		R/W	8	0x06

[Note]
Word access is available for the SFRs(Specific Function Registers) that have the word symbol. Specify the even address to word-access the registers.

7.2.2 Low Speed Time Base Counter Register (LTBR)

Address: 0xF060
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								LTBR							
Bit symbol	-	-	-	-	-	-	-	-	T1HZ	T2HZ	T4HZ	T8HZ	T16HZ	T32HZ	T64HZ	T128HZ
Access type	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

The low speed time base counter register (LTBR) is a specific function register (SFR) to read the value of the time base counter.
If LTBR is written, the content of it becomes "0." The written data is invalid.
LTBR is initialized to “0x00” at the system reset.

[Note]
- A time base counter interrupt may occur depending on a write timing to the LTBR. See the program example for initializing described in “7.3.1 Operation of the Low-speed Time Base Counter”.
- T128HZ ~ T1HZ signals have “0” level in the first half cycle and “1” level in the last half. For example, T1HZ signal gets reset to “0” by writing any data to LTBR and it get to “1” about 0.5sec later and returns to “1” about 1sec later from the reset. The low-speed time base counter interrupt occurs at the falling edge (“1” to “0”) of the signal. See “Time base counter interrupt timing and reset timing of reset by writing to LTBR” for details about the waveform of signal.

7.2.3 Low Speed Time Base Register Control Register (LTBCCON)

Address: 0xF062
Access: R/W
Access size: 8 bits
Initial value: 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								LTBCCON							
Bit symbol	-	-	-	-	-	-	-	-	-	LTCO	-	-	-	-	-	TBRUN
Access type	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	1

The low speed time base counter control register (LTBCCON) is a specific function register (SFR) to control the function of the time base counter.

- Description of bits**TBRUN** (Bit 0)
  TBRUN is a bit to control the start and stop of the time base counter.

TBRUN	Description
0	Time base counter stop
1	Time base counter start/in operation (initial value)

- **LTCO** (Bit 6)
  LTCO is a bit to select the clock to output from the time base clock output pin (TBCOUT).
  To output the clock from TBCOUT, it is necessary to set the second to octic function output of GPIO.

LTCO	Description
0	T1HZ selected (initial value)
1	T2HZ selected

[Note]
- It takes max. two clocks of the low-speed clock (LSCLK) to start or stop the operation after writing data to TBRUN bit.
- When using the on-chip debug function, the TBCOUT1 output stops during break status even if the item "Low-speed Time Base Counter" is chosen for continuing the operation during the break status on the debugger.

7.2.4 Low Speed Time Base Counter Interrupt Selection Register (LTBINT)

Address: 0xF068
Access: R/W
Access size: 8/16 bits
Initial value: 0x0630

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	LTBINT															
Byte symbol	LTBINTH								LTBINTL							
Bit symbol	-	-	-	-	-	LTI2S2	LTI2S1	LTI2S0	-	LTI1S2	LTI1S1	LTI1S0	-	LTI0S2	LTI0S1	LTI0S0
Access type	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0

The low speed time base counter interrupt selection register (LTBINT) is a specific function register (SFR) to specify the low-speed time base clock to be used as an interrupt signal.

Description of bits

- **LTI0S2 to LTI0S0**(Bit 2 to 0)
  These are bits to select the signal to be assigned to the time base counter interrupt 0 (TBCINT0). The initial value is T128HZ.

- **LTI1S2 to LTI1S0** (Bit 6 to 4)
  These are bits to select the signal to be assigned to the time base counter interrupt 1 (TBCINT1). The initial value is T16HZ.

- **LTI2S2 to LTI2S0** (Bit 10 to 8)
  These are bits to select the signal to be assigned to the time base counter interrupt 2 (TBCINT2). The initial value is T2HZ.

(n = 2, 1, 0)

LTInS2	LTInS1	LTInS0	Assigned clock
0	0	0	T128HZ
0	0	1	T64HZ
0	1	0	T32HZ
0	1	1	T16HZ
1	0	0	T8HZ
1	0	1	T4HZ
1	1	0	T2HZ
1	1	1	T1HZ

[Note]
A time base counter interrupt may occur depending on a write timing to the LTBINTL or LTLBINTH. See the program example for initializing described in “7.3.1 Operation of the Low-speed Time Base Counter”.

7.3 Description of Operation

7.3.1 Operation of Low Speed Time Base Counter

After the ssytem reset released, the low speed time base counter (LTBC) starts counting up from 0x0000at the falling edge of LSCLK to generate the signals of T32KHZ ~ T1HZ. Three interrupts are selectable among eight periodical interrupt request (T128 HZ, T64HZ, T32HZ, T16HZ, T8HZ, T4HZ, T2HZ and T1HZ).
The signals of T128HZ to T1HZ of LTBC can be read from the low speed time base counter register (LTBR).
The low speed time base counter interrupts are required at the falling edges of the signals (the initial values are T128HZ, T16HZ, and T2HZ) assigned in the low speed time base counter interrupt selection register (LTBINTL and LTBINTH).
When selecting the interrupts by the LTBINTLand LTBINTH, low speed time base counter interrupts (LTBCnINT, n=0~2) may occur depending on the write timing to the LTBINTL and LTBINTH.
Therefore, select the interrupt request (change the LTBINTL and LT BINTH) after disabling the interrupt and clear the request bits (QLTBCn, n=0~2).

Figure 7-2 shows an example of program for changing the selection of low speed time base counter interrupts.
After writing to LTBINTL and LTBINTH, it takes a time of one system clock for getting the low speed time base counter n interrupt request flag (QLTBCn, n=0-2) to "1". Therefore, place one NOP instruction after writing to the LTBR to surely clear the interrupt request flag.

```
ELTBC0 = 0;          // Disable LTBC0 interrupt
ELTBC1 = 0;          // Disable LTBC1 interrupt
ELTBC2 = 0;          // Disable LTBC2 interrupt
LTBINTL = 0x41;      // Change selection of interrupts
LTBINTH = 0x07;      // Change selection of interrupts
__asm(“NOP”);        // Wait
QLTBC0 = 0;          // Clear QLTBC0
QLTBC1 = 0;          // Clear QLTBC1
QLTBC2 = 0;          // Clear QLTBC2
ELTBC0 = 0;          // Enable LTBC0 interrupt
ELTBC1 = 0;          // Enable LTBC1 interrupt
ELTBC2 = 0;          // Enable LTBC2 interrupt
```

Figure 7-2 Program example for changing the selection of low speed time base counter interrupts

T128HZ~T1HZ signals get reset to “0” by writing any data to LTBR, and at that time, T32KHZ~T256HZ signals do not get reset.
When selecting the interrupts by the LTBINTLand LTBINTH, the low speed time base counter interrupts (LTBCnINT, n=0~2) may occur depending on the write timing to the LTBR.
Therefore, select the interrupt request (change the LTBINTL and LT BINTH) after disabling the interrupt and clear the request bits (QLTBCn, n=0~2).

Figure 7-3 shows an example of program for changing the selection of low speed time base counter interrupts.
After writing to LTBR, it takes a time of one system clock for getting the time base counter n interrupt request flag (QLTBCn, n=0-2) to "1". Therefore, place one NOP instruction after writing to the LTBR to surely clear the interrupt request flag.

```
__DI();              // Disable interrupts (MIE=0)
LTBR = 0x00;         // Reset the LRBR
__asm(“NOP”);        // Wait
QLTBC0 = 0;          // Clear QLTBC0
QLTBC1 = 0;          // Clear QLTBC1
QLTBC2 = 0;          // Clear QLTBC2
__EI();              // Eisable interrupts (MIE=1)
```

Figure 7-3 Program example for initializing the LTBR

Figure 7-4 shows the timing of the low speed time base counter interrupt generation by writing to LTBR (when assigned to T128HZ, T16HZ, and T2HZ by the time base counter interrupt selection register LTBINTL and LTBINTH).

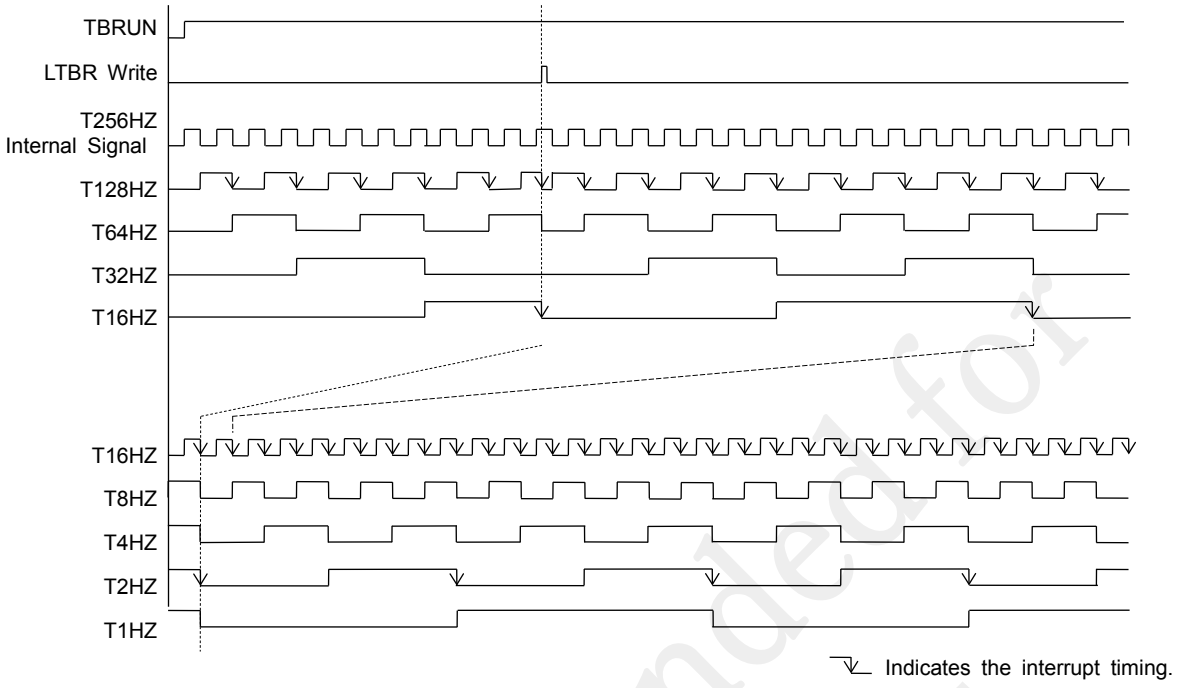


Figure 7-4 Low speed time base counter interrupt timing and reset timing of reset by writing to LTBR

[Note]
- The time from when writing the LTBR register until when the 1st low-speed counter interrupt request generates, is not guranteed. Measure times with reference to the interval of interrupt occurrence.
- It takes max. one clock of the low-speed clock (LSCLK) to reset the time base counter after writing the LTBR register.

Chapter 8 16-Bit Timer

8. 16-Bit Timer

8.1 General Description

ML62Q1000 series has 16-bit timers that count the clock based on LSCLK (low-speed clock) or HSCLK (high-speed clock).
See “Table 1-2 Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

8.1.1 Features

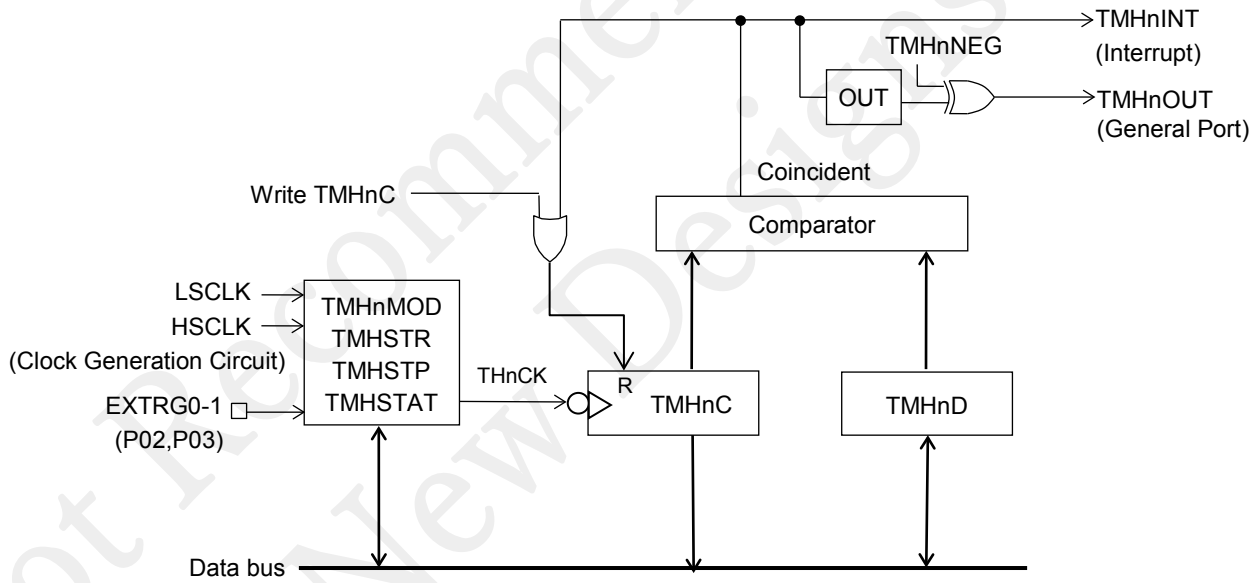
- Channel: Max.6ch

Package type	SSOP16 WQFN16 TSSOP20	WQFN24 TQFP32
Number of channel	4	6

- Repeat mode or one-shot mode is selectable as the operation mode
- 1 to 128 dividing of the LSCLK/HSCLK clock or the external trigger is selectable for the timer clock
- A toggled signal can be driven out of TMHnOUT pin (n = 0 to 5) every counter overflow.
- The output logic of TMHnOUT is selectable (Possitive or Negative).
- A timer interrupt (TMHnINT) is generated when the value of the timer counter register (TMHnC) value coincides with that of the timer data register (TMHnD). (n = 0 to 5)
- One channel of 16-bit timer is configurable as two channels of 8-bit timer

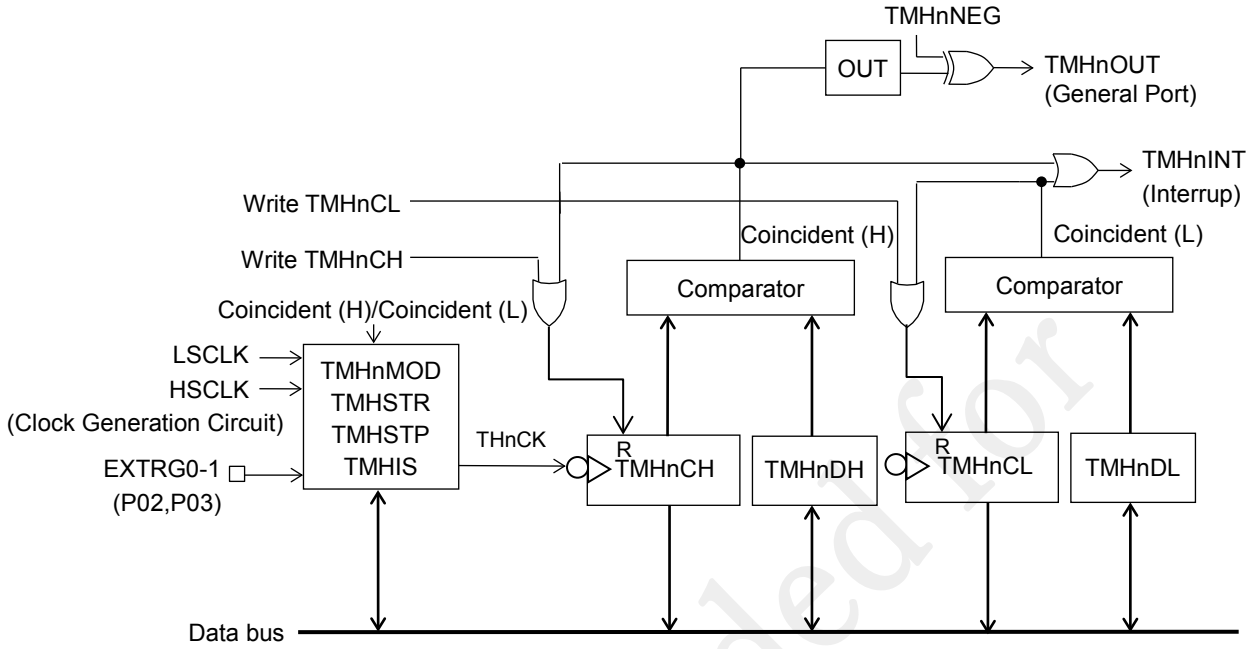
8.1.2 Configuration

Figure 8-1 shows the 16-bit timer configuration and Figure 8-2 the 8-bit timer configuration.



TMHnMOD : 16-bit timer mode register
TMHSTR : 16-bit timer start register
TMHSTP : 16-bit timer stop register
TMHSTAT : 16-bit timer status register
TMHnD : 16-bit timer n data register
TMHnC : 16-bit timer n counter register (n = 0 to 5)

Figure 8-1 16-Bit Timer Configuration



TMHnMOD : 16-bit timer mode register
TMHSTR : 16-bit timer start register
TMHSTP : 16-bit timer stop register
TMHIS : 16-bit timer interrupt status register
TMHnDL : 16-bit timer n data register (lower side)
TMHnDH : 16-bit timer n data register (upper side)
TMHnCL : 16-bit timer n counter register (lower side)
TMHnCH : 16-bit timer n counter register (upper side) (n = 0 to 5)

Figure 8-2 8-Bit Timer Configuration

[Note]
- When a 16-bit timer is used as two channels of 8-bit timer, the same clock settings and interrupts are applied.
- For TMHnOUT (timer out) of the 8-bit timer mode, values on the upper side ("TMHnDH" and "TMHnCH") are output.

8.1.3 List of Pins

The output pins of the 16-bit timer are assigned to the second to octic functions of GPIO.
For details of pin assignment and the second to octic settings of GPIO, see Chapter 17 "GPIO."

Pin name	I/O	Description
EXTRG0-1	I	External clock input
TMHnOUT (n=0 to 5)	O	Output of 16-bit timer n When used in an 8-bit timer, output is possible only from the upper timer.

8.2 Description of Registers

8.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF300	16-bit timer 0 data register	TMH0DL	TMH0D	R/W	8/16	0xFF
0xF301		TMH0DH		R/W	8	0xFF
0xF302	16-bit timer 1 data register	TMH1DL	TMH1D	R/W	8/16	0xFF
0xF303		TMH1DH		R/W	8	0xFF
0xF304	16-bit timer 2 data register	TMH2DL	TMH2D	R/W	8/16	0xFF
0xF305		TMH2DH		R/W	8	0xFF
0xF306	16-bit timer 3 data register	TMH3DL	TMH3D	R/W	8/16	0xFF
0xF307		TMH3DH		R/W	8	0xFF
0xF308	16-bit timer 4 data register	TMH4DL	TMH4D	R/W	8/16	0xFF
0xF309		TMH4DH		R/W	8	0xFF
0xF30A	16-bit timer 5 data register	TMH5DL	TMH5D	R/W	8/16	0xFF
0xF30B		TMH5DH		R/W	8	0xFF
0xF310	16-bit timer 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF311		TMH0CH		R/W	8	0x00
0xF312	16-bit timer 1 counter register	TMH1CL	TMH1C	R/W	8/16	0x00
0xF313		TMH1CH		R/W	8	0x00
0xF314	16-bit timer 2 counter register	TMH2CL	TMH2C	R/W	8/16	0x00
0xF315		TMH2CH		R/W	8	0x00
0xF316	16-bit timer 3 counter register	TMH3CL	TMH3C	R/W	8/16	0x00
0xF317		TMH3CH		R/W	8	0x00
0xF318	16-bit timer 4 counter register	TMH4CL	TMH4C	R/W	8/16	0x00
0xF319		TMH4CH		R/W	8	0x00
0xF31A	16-bit timer 5 counter register	TMH5CL	TMH5C	R/W	8/16	0x00
0xF31B		TMH5CH		R/W	8	0x00
0xF320	16-bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF321		TMH0MODH		R/W	8	0x00
0xF322	16-bit timer 1 mode register	TMH1MODL	TMH1MOD	R/W	8/16	0x00
0xF323		TMH0MODH		R/W	8	0x00
0xF324	16-bit timer 2 mode register	TMH2MODL	TMH2MOD	R/W	8/16	0x00
0xF325		TMH0MODH		R/W	8	0x00
0xF326	16-bit timer 3 mode register	TMH3MODL	TMH3MOD	R/W	8/16	0x00
0xF327		TMH3MODH		R/W	8	0x00
0xF328	16-bit timer 4 mode register	TMH4MODL	TMH4MOD	R/W	8/16	0x00
0xF329		TMH4MODH		R/W	8	0x00
0xF32A	16-bit timer 5 mode register	TMH5MODL	TMH5MOD	R/W	8/16	0x00
0xF32B		TMH5MODH		R/W	8	0x00
0xF330	16-bit timer 0 interrupt status register	TMH0ISL	TMH0IS	R	8/16	0x00
0xF331		TMH0ISH		R	8	0x00
0xF332	16-bit timer 1 interrupt status register	TMH1ISL	TMH1IS	R	8/16	0x00
0xF333		TMH1ISH		R	8	0x00
0xF334	16-bit timer 2 interrupt status register	TMH2ISL	TMH2IS	R	8/16	0x00
0xF335		TMH2ISH		R	8	0x00
0xF336	16-bit timer 3 interrupt status register	TMH3ISL	TMH3IS	R	8/16	0x00
0xF337		TMH3ISH		R	8	0x00

0xF338	16-bit timer 4 interrupt status register	TMH4ISL	TMH4IS	R	8/16	0x00
0xF339		TMH4ISH		R	8	0x00
0xF33A	16-bit timer 5 interrupt status register	TMH5ISL	TMH5IS	R	8/16	0x00
0xF33B		TMH5ISH		R	8	0x00
0xF340	16-bit timer 0 interrupt clear register	TMH0ICL	TMH0IC	W	8/16	0x00
0xF341		TMH0ICH		W	8	0x00
0xF342	16-bit timer 1 interrupt clear register	TMH1ICL	TMH1IC	W	8/16	0x00
0xF343		TMH1ICH		W	8	0x00
0xF344	16-bit timer 2 interrupt clear register	TMH2ICL	TMH2IC	W	8/16	0x00
0xF345		TMH2ICH		W	8	0x00
0xF346	16-bit timer 3 interrupt clear register	TMH3ICL	TMH3IC	W	8/16	0x00
0xF347		TMH3ICH		W	8	0x00
0xF348	16-bit timer 4 interrupt clear register	TMH4ICL	TMH4IC	W	8/16	0x00
0xF349		TMH4ICH		W	8	0x00
0xF34A	16-bit timer 5 interrupt clear register	TMH5ICL	TMH5IC	W	8/16	0x00
0xF34B		TMH5ICH		W	8	0x00
0xF350	16-bit timer start register	TMHSTRL	TMHSTR	R/W	8/16	0x00
0xF351		TMHSTRH		R/W	8	0x00
0xF352	16-bit timer stop register	TMHSTPL	TMHSTP	R/W	8/16	0x00
0xF353		TMHSTPH		R/W	8	0x00
0xF354	16-bit timer status register	TMHSTATL	TMHSTAT	R/W	8/16	0x00
0xF355		TMHSTATH		R/W	8	0x00

[Note]
Word access is available for the SFRs(Specific Function Registers) that have the word symbol. Specify the even address to word-access the registers.

8.2.2 16-Bit Timer n Data Register (TMHnD: n = 0 to 5)

Address: 0xF300, 0xF302, 0xF304, 0xF306, 0xF308, 0xF30A
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

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

TMHnD (n = 0 to 5) is a specific function register (SFR) to set the comparison value with the 16-bit timer n counter register (TMHnC).
In the 8-bit timer mode, TMHnDL (n = 0 to 5) is compared to TMHnCL (n = 0 to 5) and TMHnDH (n = 0 to 5) is compared to TMHnCH (n = 0 to 5).

[Note]
Set TMHnD when the 16-bit timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are “0”).
When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD.
When "0x00" is written in TMHnDL/TMHnDH in the 8-bit timer mode, "0x01" is set in TMHnDL/TMHnDH.

8.2.3 16-Bit Timer n Counter Register (TMHnC: n = 0 to 5)

Address: 0xF310, 0xF312, 0xF314, 0xF316, 0xF318, 0xF31A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

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

TMHnC (n = 0 to 5) is a specific function register (SFR) that functions as a 16-bit binary counter.
When an arbitrary value is written in TMHnC in the 16-bit timer mode, the count value becomes "0x0000."
When an arbitrary value is written in TMHnCH and TMHnCL respectively in the 8-bit timer mode, the count value becomes "0x00."

8.2.4 16-Bit Timer n Mode Register (TMHnMOD: n = 0 to 5)

Address: 0xF320, 0xF322, 0xF324, 0xF326, 0xF328, 0xF32A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHnMOD															
Byte symbol	TMHnMODH								TMHnMODL							
Bit symbol	-	-	-	-	-	THnNEG	THnOST	THn8BM	-	THnDIV2	THnDIV1	THnDIV0	THnEXS	THnEX	-	THnCS
Access type	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMHnMOD (n = 0 to 5) is a specific function register (SFR) to control 16-bit timer 8.
Rewrite TMHnMOD only when 16-bit timer 8 is stopped.

Description of bits

- **THnCS** (Bit 0)
  THnCS is a bit to select the timer clock of the 16-bit timer n timer.

THnCS	Description
0	LSCLK (initial value)
1	HSCLK

- **THnEX** (Bit 2)
  THnEX is a bit to select the count clock of the 16-bit timer n.
  The timer clock and the count clock are different (same when **THnDIV[2:0]** = 0). The count clock is used for counting operation and waveform output control and the timer clock is used for sampling and edge detection of other external triggers.

THnEX	Description
0	The timer clock selected in the THnCS bit is counted with the dividing ratio selected in the THnDIV2-0 bit. (initial value)
1	The rising edge of the external trigger selected in the THnEXS bit is detected with the timer clock selected in the THnCS bit to count.

- **THnEXS** (Bit 3)
  THnEXS is a bit to select the external trigger that is the count clock of the 16-bit timer n.

THnEXS	Description
0	P02 (initial value)
1	P03

[Note]
Input the pulse for the external trigger with the width of two timer clocks or longer.

- **THnDIV2 to THnDIV0** (Bit 4 to 6)
  THnDIV2 to THnDIV0 are bits to select the count clock dividing ratio of the 16-bit timer n timer.

THnDIV2	THnDIV1	THnDIV0	Description
0	0	0	1 dividing (initial value)
0	0	1	1/2 dividing
0	1	0	1/4 dividing
0	1	1	1/8 dividing
1	0	0	1/16 dividing
1	0	1	1/32 dividing
1	1	0	1/64 dividing
1	1	1	1/128 dividing

- **THn8BM** (Bit 8)
  THn8BM is a bit to select whether one 16-bit timer or two channels of 8-bit timer.

THn8BM	Description
0	16-bit timer mode (initial value)
1	8-bit timer mode

- **THnOST** (Bit 9)
  THnOST is a bit to select the operation mode of the 16-bit timer n.

THnOST	Description
0	Repeat timer mode (initial value)
1	One-shot timer mode

- **THnNEG** (Bit 10)
  THnNEG is a bit to select the output polarity of timer out (TMHnOUT).

THnNEG	Description
0	Positive logic (initial value)
1	Negative logic

[Note]
Set TMHnMOD when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are “0”).
If it is changed while it is operating, the operation is not guaranteed.
In the 8-bit timer mode, the operation mode specified by THnCS0 to 1, THnDIV2 to 0 and THnOST are common for both two channels.

8.2.5 16-Bit Timer n Interrupt Status Register (TMHnIS: n = 0 to 5)

Address: 0xF330, 0xF332, 0xF334, 0xF336, 0xF338, 0xF33A
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 symbol	TMHnIS															
Byte symbol	TMHnISH								TMHnISL							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	THnHIS	THnLIS
Access type	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

TMHnIS is a specific function register (SFR) to indicate the status of the interrupt used in the 8-bit timer mode. It is possible to check which interrupt was output in the 8-bit timer mode. It is fixed to 0 in the 16-bit timer mode. When the interrupt status on the upper or lower side of the same channel is "0", the next interrupts on the upper and lower sides are not output. Clear it by writing "1" to TMHnIC.

Description of setting value

Setting value	Description
0	8-bit timer interrupt does not occur (initial value)
1	8-bit timer interrupt occurs

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 1	THnHIS	Interrupt on upper side of 16-bit timer n
Bit 0	THnLIS	Interrupt on lower side of 16-bit timer n

8.2.6 16-Bit Timer n Interrupt Clear Register (TMHnIC: n = 0 to 5)

Address: 0xF340, 0xF342, 0xF344, 0xF346, 0xF348, 0xF34A
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 symbol	TMHnIC															
Byte symbol	TMHnICH								TMHnICL							
Bit symbol	-	-	-	-	-	-	-	-	THnIR	-	-	-	-	-	THnHIC	THnLIC
Access type	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

TMHnIC is a write-only specific function register (SFR) to clear the status of the interrupt used in the 8-bit timer mode. When writing 1 to this bit, the target interrupt status is cleared. When reading it, 0x0000 is always read.
This is not used in the 16-bit mode.

- **THnLIC** (Bit 0)
  This is a bit to clear the lower side interrupt of the 8-bit timer n.

THnLIC	Description
0	Remain the lower side interrupt status of the 16-bit timer n (initial value)
1	Clear the lower side interrupt status of the 16-bit timer n to 0 (initial value)

- **THnHIC** (Bit 1)
  This is a bit to clear the upper side interrupt of the 8-bit timer n.

THnHIC	Description
0	Remain the upper side interrupt status of the 16-bit timer n (initial value)
1	Clear the upper side interrupt status of the 16-bit timer n to 0

- **THnIR** (Bit 7)
  This is a bit to generate the unprocessed (when an interrupt request not to be cleared remains in the TMHnIS register) interrupt request of the 8-bit timer n. To generate the unprocessed interrupt request, write "1" before exiting from the interrupt vector.

THnIR	Description
0	Unprocessed interrupt request not generated (initial value)
1	Unprocessed interrupt request generated

8.2.7 16-Bit Timer Start Register (TMHSTR)

Address: 0xF350
Access: W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHSTR															
Byte symbol	TMHSTRH								TMHSTRL							
Bit symbol	-	-	TH5HRUN	TH4HRUN	TH3HRUN	TH2HRUN	TH1HRUN	TH0HRUN	-	-	TH5RUN	TH4RUN	TH3RUN	TH2RUN	TH1RUN	TH0RUN
Access type	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

TMHSTRL is a specific function register (SFR) to control counting start of the 16-bit timer n (0 to 5).
TMHSTRH is used in the 8-bit timer mode.
TMHSTR is a write-only register to control the operation start of the timer. Operation of the timer is stopped by TMHSTP.

Description of setting value

Setting value	Description
0	Remain the current status (initial value)
1	Start counting

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 13	TH5HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 5
Bit 12	TH4HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 4
Bit 11	TH3HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 3
Bit 10	TH2HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 2
Bit 9	TH1HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 1
Bit 8	TH0HRUN	In the 8-bit timer mode: Upper side timer start control of 16-bit timer 0
Bit 5	TH5RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 5 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 5
Bit 4	TH4RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 4 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 4
Bit 3	TH3RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 3 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 3
Bit 2	TH2RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 2 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 2
Bit 1	TH1RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 1 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 1
Bit 0	TH0RUN	In the 16-bit timer mode: Timer start control of 16-bit timer 0 In the 8-bit timer mode: Lower side timer start control of 16-bit timer 0

[Note]
In the 16-bit timer mode, bit 13 to 8(THnHRUN bit(n=0 to 5)) are not used. Writing “1” to the bis are ignored.
Set TMHSTR when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are “0”).

8.2.8 16-Bit Timer Stop Register (TMHSTP)

Address: 0xF352
Access: W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHSTP															
Byte symbol	TMHSTPH								TMHSTPL							
Bit symbol	-	-	TH5HSTP	TH4HSTP	TH3HSTP	TH2HSTP	TH1HSTP	TH0HSTP	-	-	TH5STP	TH4STP	TH3STP	TH2STP	TH1STP	TH0STP
Access type	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

TMHSTPL is a specific function register (SFR) to control counting stop of the 16-bit timer n (n = 0 to 5).
TMHSTPH is used in the 8-bit timer mode.

Description of setting value

Setting value	Description
0	Remain current status (initial value)
1	Stop counting

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 13	TH5HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 5
Bit 12	TH4HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 4
Bit 11	TH3HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 3
Bit 10	TH2HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 2
Bit 9	TH1HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 1
Bit 8	TH0HSTP	In the 8-bit timer mode: Upper side timer stop control of 16-bit timer 0
Bit 5	TH5STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 5 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 5
Bit 4	TH4STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 4 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 4
Bit 3	TH3STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 3 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 3
Bit 2	TH2STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 2 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 2
Bit 1	TH1STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 1 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 1
Bit 0	TH0STP	In the 16-bit timer mode: Timer stop control of 16-bit timer 0 In the 8-bit timer mode: Lower side timer stop control of 16-bit timer 0

[Note]
- In the 16-bit timer mode, bit 15 to 8 are not used. They are enabled when "1" is written.
- Set TMHSTR when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are "1").

8.2.9 16-Bit Timer Status Register (TMHSTAT)

Address: 0xF354
Access: R
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHSTAT															
Byte symbol	TMHSTATH								TMHSTATL							
Bit symbol	-	-	TH5HSTAT	TH4HSTAT	TH3HSTAT	TH2HSTAT	TH1HSTAT	TH0HSTAT	-	-	TH5STAT	TH4STAT	TH3STAT	TH2STAT	TH1STAT	TH0STAT
Access type	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

TMHSTATL is a specific function register (SFR) to indicate the status of the 16-bit timer n (n = 0 to 5).
TMHSTATH is used in the 8-bit timer mode. It is fixed to 0 in the 16-bit timer mode.

Description of setting value

Setting value	Description
0	Counting stopped (Initial value)
1	Counting in progress

Corresponding block

Bit	Bit symbol name	Corresponding interrupt
Bit 13	TH5HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 5
Bit 12	TH4HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 4
Bit 11	TH3HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 3
Bit 10	TH2HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 2
Bit 9	TH1HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 1
Bit 8	TH0HSTAT	In the 8-bit timer mode: Upper side timer status of 16-bit timer 0
Bit 5	TH5STAT	In the 16-bit timer mode: Timer status of 16-bit timer 5 In the 8-bit timer mode: Lower side timer status of 16-bit timer 5
Bit 4	TH4STAT	In the 16-bit timer mode: Timer status of 16-bit timer 4 In the 8-bit timer mode: Lower side timer status of 16-bit timer 4
Bit 3	TH3STAT	In the 16-bit timer mode: Timer status of 16-bit timer 3 In the 8-bit timer mode: Lower side timer status of 16-bit timer 3
Bit 2	TH2STAT	In the 16-bit timer mode: Timer status of 16-bit timer 2 In the 8-bit timer mode: Lower side timer status of 16-bit timer 2
Bit 1	TH1STAT	In the 16-bit timer mode: Timer status of 16-bit timer 1 In the 8-bit timer mode: Lower side timer status of 16-bit timer 1
Bit 0	TH0STAT	In the 16-bit timer mode: Timer status of 16-bit timer 0 In the 8-bit timer mode: Lower side timer status of 16-bit timer 0

8.3 Description of Operation

8.3.1 Basic Operation of 16-Bit Timer

In the 16-bit timer n counter (TMHnC, n = 0 to 5), when "1" is written in the THnRUN bit of the 16-bit timer start register (TMHSTR) , THnSTAT enters the operation status of "1" at the first rising edge of the timer n clock (THnCK) selected in the 16-bit timer n mode register (TMHnMOD) and counting is started at the second rising edge.
When the count value of TMHnC coincides with the value of the timer n data register (TMHnD), the timer n interrupt (TMnINT) is generated at the next THnCK rising edge and TMHnC is reset to "0x0000" at the same time.
Counting is continued in the repeat timer mode and stopped in the one-shot timer mode.

When the THnSTP bit of the 16-bit timer stop register (TMHSTP) is set to "1", TMHnC stops counting at the first rising edge of THnCK and THnSTAT becomes "0". When the THnRUN bit is set to "0" again, TMHnC restarts counting from the value at the time of stop. To initialize THMnC to "0x0000", write an arbitrary value to TMHnC.

The timer interrupt period (T_{TMI}) is expressed by the following equation.

$$T_{TMI} = \frac{TMHnD + 1}{THnCK\ (Hz)} \quad (n = 0 \text{ to } 7)$$

TMHnD : 16-bit timer n data register (TMHnD) setting value (0x0001 to 0xFFFF)
THnCK : Clock frequency selected by the 16-bit timer n mode register (TMHnMOD)

When the THnRUN bit is set to "1", counting is started by synchronizing with THnCK, so an error of up to 1 clock of THnCK is generated for the first timer interrupt.
Subsequent timer interrupt periods are constant. Similarly, when the THnSTP bit is set to "1", counting is stopped by synchronizing with THnCK, so a timer n interrupt (TMHnINT) may be generated depending on the stop timing.

Figure 8-3 shows the operation waveforms in the repeat timer mode and Figure 8-4 shows the operation waveforms in the one-shot timer mode.

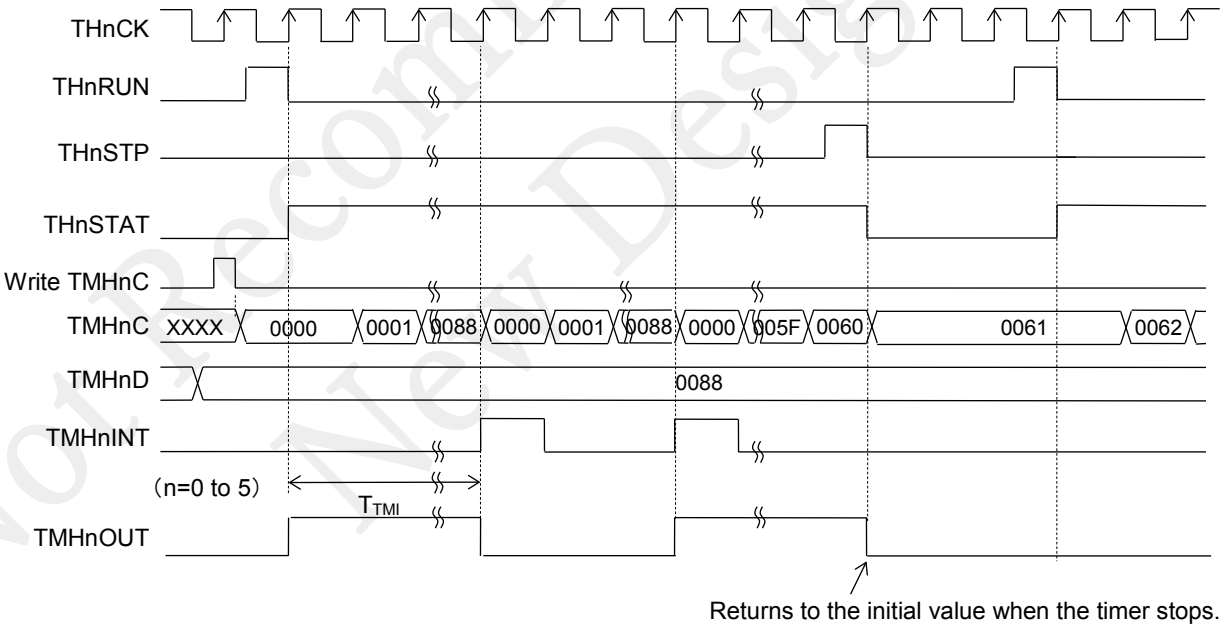


Figure 8-3 Operation Waveforms of Repeat Timer

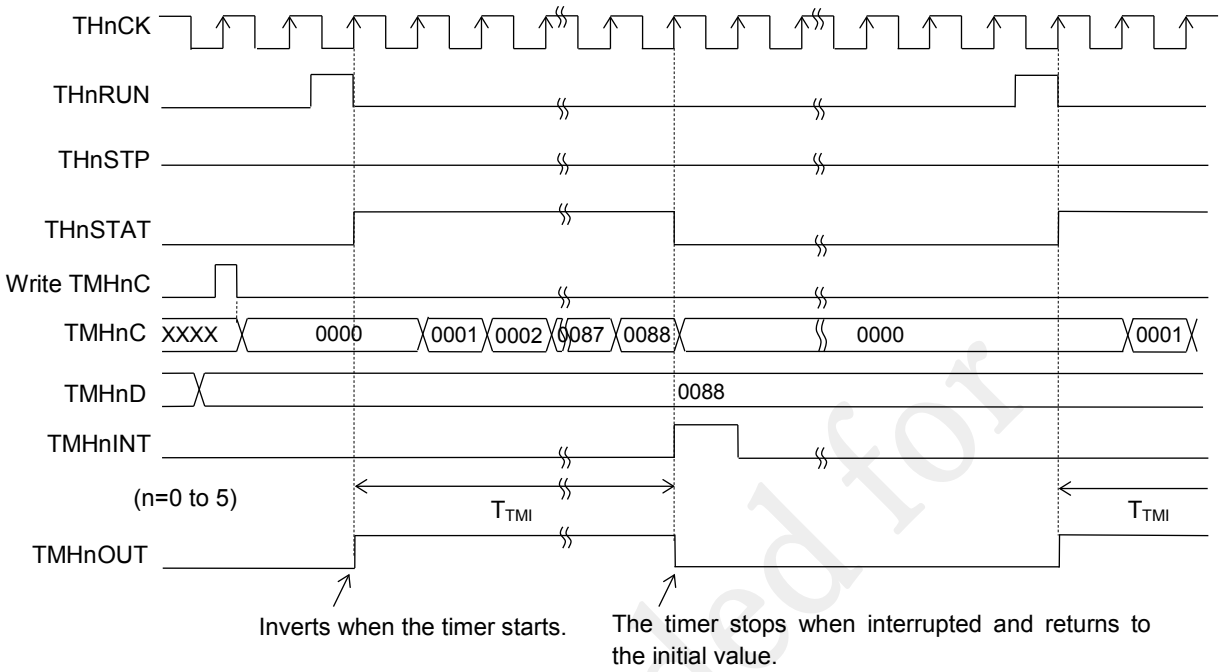


Figure 8-4 Operation Waveforms of One-Shot Timer

8.3.2 Setting Example of Repeat Timer Mode

Figure 8-5 shows a setting example to generate an interrupt every 100 ms in the repeat timer mode by using 16-bit timer 0.

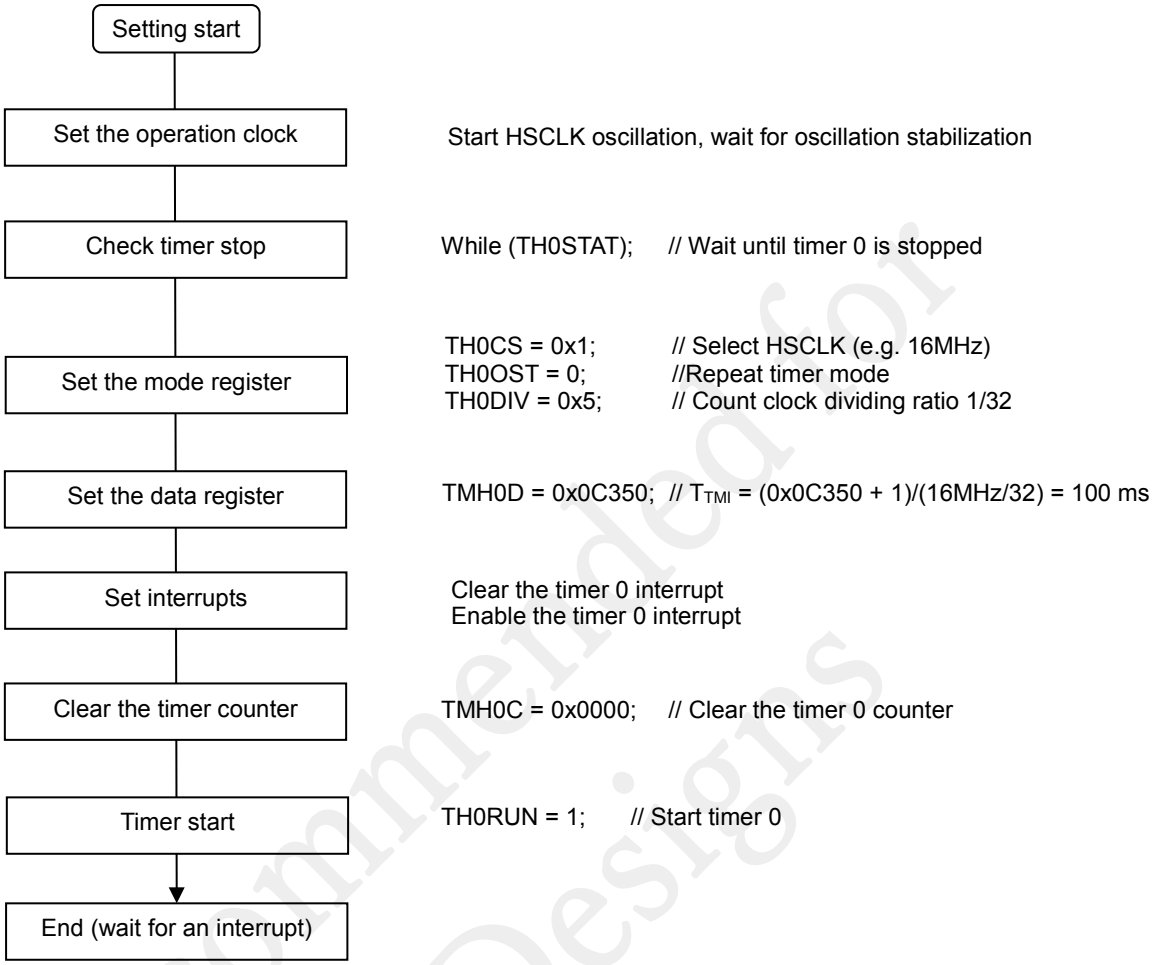


Figure 8-5 Setting Example of Repeat Timer Mode

8.3.3 External Input Count Timing

When the external input is selected for the count clock, the input pulse is sampled by the timer clock detecting the rising edge and counting up the timer.
Figure 8-6 shows the count timing.

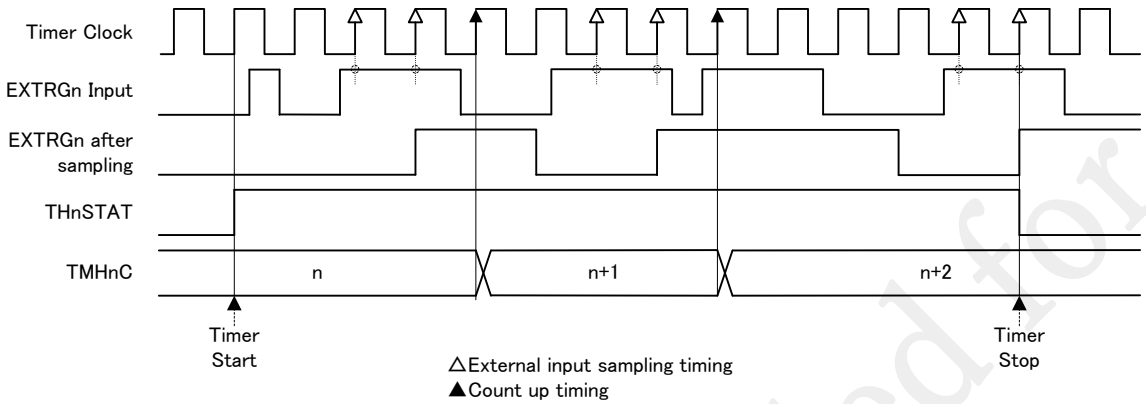


Figure 8-6 External Input Count Timing

Figure 8-7 shows the setting example when using the external input.

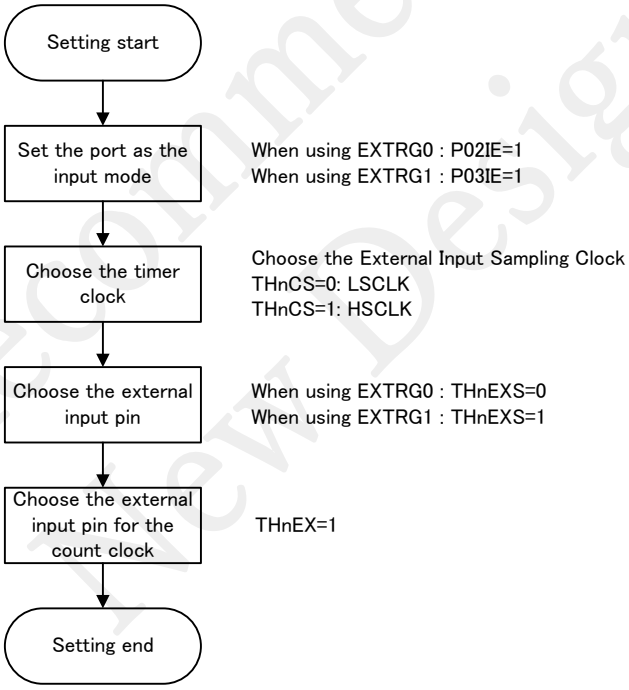


Figure 8-7 External Input Count Setting Example

[Note]
- The external input pluse with width shorter than two clocks of timer clock may be ignored. Input the external clock signal with the two clocks width of timer clock or longer.
- The external input signal for the 16bit Timer (EXTRGn) comes through the sampling controller in the external interrupt function. The sampling for the external interrupt is selectable to use or not. See Chapter 18 “External Interrupt Control Circuit”.

Chapter 9 Functional Timer (FTM)

9. Functional Timer (FTM)

9.1 General Description

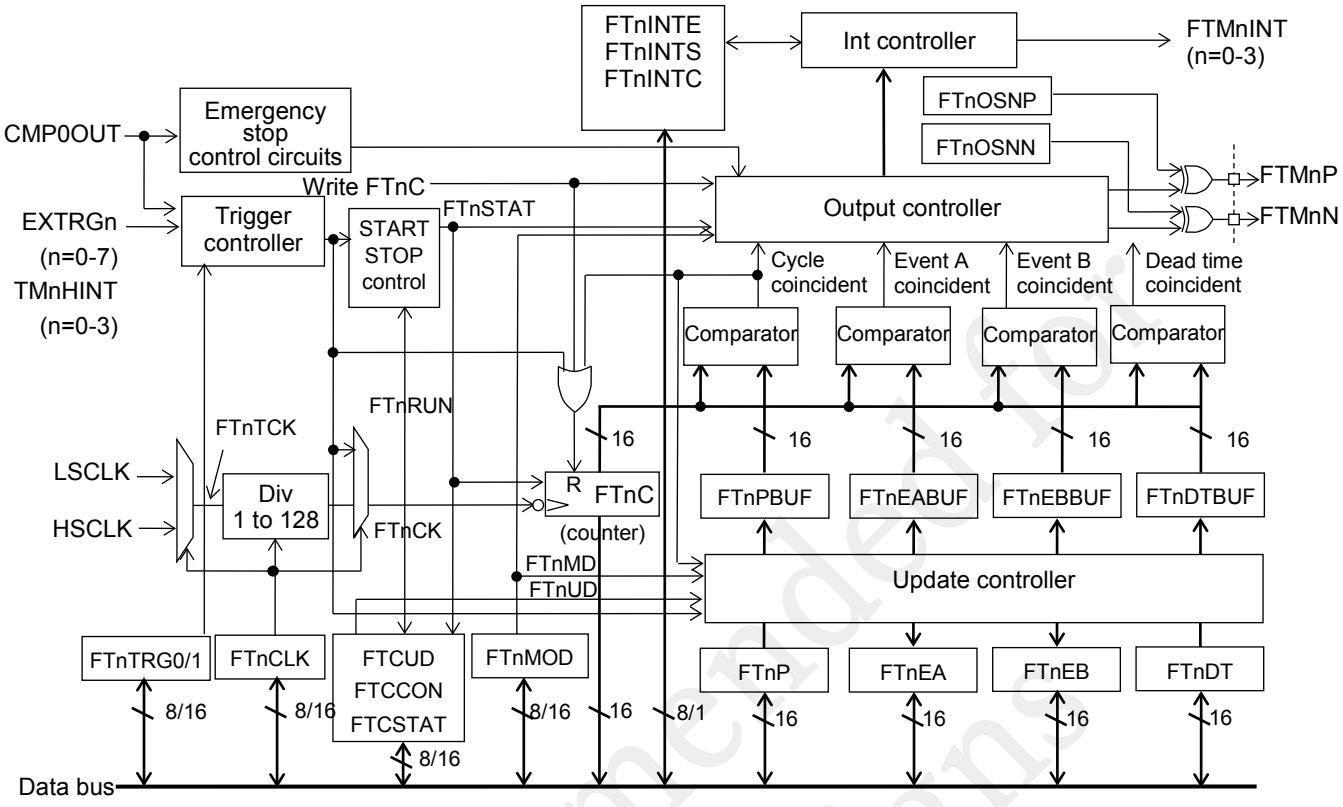
ML62Q1000 series has functional timers equipped with the timer function, capture function and PWM function.
See “Table 1-2 Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

9.1.1 Features

- Equipped with the timer/capture/PWM functions using a 16-bit counter
- For the count clock, 1 to 128 dividing of the LSCLK/HSCLK clock and the external trigger can be selected
- The timer output signal can be switched between the positive and negative logics
- Duty interrupt and coincident interrupt with the setting value as well as the cyclic interrupt generated
- Equipped with one-shot mode
- Start/stop/clear control of the timer is possible with the event trigger (external pin input interrupt, timer interrupt request) (however, the minimum pulse width of pin input is timer clock 3φ)
- Emergency stop and emergency stop interrupt by an external input
- Two types of PWM output with the same period and different duties, and complementary PWM output with the dead time
- Input signal duty/cycle measurement by the capture function
- Interrupt source to be notified can be set

9.1.2 Configuration

Figure 9-1 shows the configuration of the FTM circuit.



FTnP	: FTMn cycle register	FTnPBUF	: FTMn cycle buffer
FTnEA/B	: FTMn event A/B register	FTnEA/BBUF	: FTMn event A/B buffer
FTnDT	: FTMn dead time register	FTnDTBUF	: FTMn dead time buffer
FTnC	: FTMn counter register	FTnMOD	: FTMn mode register
FTCCON	: FTM common control register	FTnCLK	: FTMn clock register
FTnTRG0/1	: FTMn trigger register 0,1	FTCUD	: FTM common update register
FTnINTE/S/C	: FTMn interrupt enable/status/clear register		
FTCSTR/STP/STAT	: FTM common start/stop/status register		
FTnTCK (n=0-3)	:Timer clock	FTnCK	:Count clock

Figure 9-1 Configuration of Circuit

9.1.3 List of Pins

Pin name	I/O	Function
EXTRG0-7	I	External clock input
FTMnP(n=0-3)	O	Timer output
FTMnN(n=0-3)	O	Timer output

9.2 Description of Registers

9.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF400	FTM0 cycle register	FT0PL	FT0P	R/W	8/16	0xFF
0xF401		FT0PH		R/W	8	0xFF
0xF402	FTM1 cycle register	FT1PL	FT1P	R/W	8/16	0xFF
0xF403		FT1PH		R/W	8	0xFF
0xF404	FTM2 cycle register	FT2PL	FT2P	R/W	8/16	0xFF
0xF405		FT2PH		R/W	8	0xFF
0xF406	FTM3 cycle register	FT3PL	FT3P	R/W	8/16	0xFF
0xF407		FT3PH		R/W	8	0xFF
0xF408～ 0xF40F	Reserved register	-	-	-	-	-
0xF410	FTM0 event A register	FT0EAL	FT0EA	R/W	8/16	0x00
0xF411		FT0EAH		R/W	8	0x00
0xF412	FTM1 event A register	FT1EAL	FT1EA	R/W	8/16	0x00
0xF413		FT1EAH		R/W	8	0x00
0xF414	FTM2 event A register	FT2EAL	FT2EA	R/W	8/16	0x00
0xF415		FT2EAH		R/W	8	0x00
0xF416	FTM3 event A register	FT3EAL	FT3EA	R/W	8/16	0x00
0xF417		FT3EAH		R/W	8	0x00
0xF418～ 0xF41F	Reserved register	-	-	-	-	-
0xF420x	FTM0 event B register	FT0EBL	FT0EB	R/W	8/16	0x00
0xF421		FT0EBH		R/W	8	0x00
0xF422	FTM1 event B register	FT1EBL	FT1EB	R/W	8/16	0x00
0xF423		FT1EBH		R/W	8	0x00
0xF424	FTM2 event B register	FT2EBL	FT2EB	R/W	8/16	0x00
0xF425		FT2EBH		R/W	8	0x00
0xF426	FTM3 event B register	FT3EBL	FT3EB	R/W	8/16	0x00
0xF427		FT3EBH		R/W	8	0x00
0xF428～ 0xF42F	Reserved register	-	-	-	-	-
0xF430	FTM0 dead time register	FT0DTL	FT0DT	R/W	8/16	0x00
0xF431		FT0DTH		R/W	8	0x00
0xF432	FTM1 dead time register	FT1DTL	FT1DT	R/W	8/16	0x00
0xF433		FT1DTH		R/W	8	0x00
0xF434	FTM2 dead time register	FT2DTL	FT2DT	R/W	8/16	0x00
0xF435		FT2DTH		R/W	8	0x00
0xF436x	FTM3 dead time register	FT3DTL	FT3DT	R/W	8/16	0x00
0xF437		FT3DTH		R/W	8	0x00
0xF438～ 0xF43F	Reserved register	-	-	-	-	-
0xF440	FTM0 counter register	FT0CL	FT0C	R/W	8/16	0x00
0xF441		FT0CH		R/W	8	0x00
0xF442	FTM1 counter register	FT1CL	FT1C	R/W	8/16	0x00
0xF443		FT1CH		R/W	8	0x00

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF444	FTM2 counter register	FT2CL	FT2C	R/W	8/16	0x00
0xF445		FT2CH		R/W	8	0x00
0xF446	FTM3 counter register	FT3CL	FT3C	R/W	8/16	0x00
0xF447		FT3CH		R/W	8	0x00
0xF448～ 0xF44F	Reserved register	-	-	-	-	-
0xF450	FTM0 status register	FT0STAT	-	R/W	8	0x30
0xF451	Reserved register	-	-	-	-	-
0xF452	FTM1 status register	FT1STAT	-	R/W	8	0x30
0xF453	Reserved register	-	-	-	-	-
0xF454	FTM2 status register	FT2STAT	-	R/W	8	0x30
0xF455	Reserved register	-	-	-	-	-
0xF456	FTM3 status register	FT3STAT	-	R/W	8	0x30
0xF457	Reserved register	-	-	-	-	-
0xF458～ 0xF45F	Reserved register	-	-	-	-	-
0xF460	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF461		FT0MODH		R/W	8	0x00
0xF462	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF463		FT1MODH		R/W	8	0x00
0xF464	FTM2 mode register	FT2MODL	FT2MOD	R/W	8/16	0x00
0xF465		FT2MODH		R/W	8	0x00
0xF466	FTM3 mode register	FT3MODL	FT3MOD	R/W	8/16	0x00
0xF467		FT3MODH		R/W	8	0x00
0xF468～ 0xF46F	Reserved register	-	-	-	-	-
0xF470	FTM0 clock register	FT0CLKL	FT0CLK	R/W	8/16	0x00
0xF471		FT0CLKH		R/W	8	0x00
0xF472	FTM1 clock register	FT1CLKL	FT1CLK	R/W	8/16	0x00
0xF473		FT1CLKH		R/W	8	0x00
0xF474	FTM2 clock register	FT2CLKL	FT2CLK	R/W	8/16	0x00
0xF475		FT2CLKH		R/W	8	0x00
0xF476	FTM3 clock register	FT3CLKL	FT3CLK	R/W	8/16	0x00
0xF477		FT3CLKH		R/W	8	0x00
0xF478～ 0xF47F	Reserved register	-	-	-	-	-
0xF480	FTM0 trigger register 0	FT0TRG0L	FT0TRG0	R/W	8/16	0x00
0xF481		FT0TRG0H		R/W	8	0x00
0xF482	FTM1 trigger register 0	FT1TRG0L	FT1TRG0	R/W	8/16	0x00
0xF483		FT1TRG0H		R/W	8	0x00
0xF484	FTM2 trigger register 0	FT2TRG0L	FT2TRG0	R/W	8/16	0x00
0xF485		FT2TRG0H		R/W	8	0x00
0xF486	FTM3 trigger register 0	FT3TRG0L	FT3TRG0	R/W	8/16	0x00
0xF487		FT3TRG0H		R/W	8	0x00
0xF488～ 0xF48F	Reserved register	-	-	-	-	-
0xF490	FTM0 trigger register 1	FT0TRG1L	FT0TRG1	R/W	8/16	0x00
0xF491		FT0TRG1H		R/W	8	0x00

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF492	FTM1 trigger register 1	FT1TRG1L	FT1TRG1	R/W	8/16	0x00
0xF493		FT1TRG1H		R/W	8	0x00
0xF494	FTM2 trigger register 1	FT2TRG1L	FT2TRG1	R/W	8/16	0x00
0xF495		FT2TRG1H		R/W	8	0x00
0xF496	FTM3 trigger register 1	FT3TRG1L	FT3TRG1	R/W	8/16	0x00
0xF497		FT3TRG1H		R/W	8	0x00
0xF498～ 0xF49F	Reserved register	-	-	-	-	-
0xF4A0	FTM0 interrupt enable register	FT0INTEL	FT0INTE	R/W	8/16	0x00
0xF4A1		FT0INTEH		R/W	8	0x00
0xF4A2	FTM1 interrupt enable register	FT1INTEL	FT1INTE	R/W	8/16	0x00
0xF4A3		FT1INTEH		R/W	8	0x00
0xF4A4	FTM2 interrupt enable register	FT2INTEL	FT2INTE	R/W	8/16	0x00
0xF4A5		FT2INTEH		R/W	8	0x00
0xF4A6	FTM3 interrupt enable register	FT3INTEL	FT3INTE	R/W	8/16	0x00
0xF4A7		FT3INTEH		R/W	8	0x00
0xF4A8～ 0xF4AF	Reserved register	-	-	-	-	-
0xF4B0	FTM0 interrupt status register	FT0INTSL	FT0INTS	R	8/16	0x00
0xF4B1		FT0INTSH		R	8	0x00
0xF4B2	FTM1 interrupt status register	FT1INTSL	FT1INTS	R	8/16	0x00
0xF4B3		FT1INTSH		R	8	0x00
0xF4B4	FTM2 interrupt status register	FT2INTSL	FT2INTS	R	8/16	0x00
0xF4B5		FT2INTSH		R	8	0x00
0xF4B6	FTM3 interrupt status register	FT3INTSL	FT3INTS	R	8/16	0x00
0xF4B7		FT3INTSH		R	8	0x00
0xF4B8～ 0xF4BF	Reserved register	-	-	-	-	-
0xF4C0	FTM0 interrupt clear register	FT0INTCL	FT0INTC	W	8/16	0x00
0xF4C1		FT0INTCH		W	8	0x00
0xF4C2	FTM1 interrupt clear register	FT1INTCL	FT1INTC	W	8/16	0x00
0xF4C3		FT1INTCH		W	8	0x00
0xF4C4	FTM2 interrupt clear register	FT2INTCL	FT2INTC	W	8/16	0x00
0xF4C5		FT2INTCH		W	8	0x00
0xF4C6	FTM3 interrupt clear register	FT3INTCL	FT3INTC	W	8/16	0x00
0xF4C7		FT3INTCH		W	8	0x00
0xF4C8～ 0xF4EF	Reserved register	-	-	-	-	-
0xF4F0	FTM common update register	FTCUD	-	W	8	0x00
0xF4F1	Reserved register	-	-	-	-	-
0xF4F2	FTM common control register L	FTCCONL	FTCCON	W/R	8/16	0x00
0xF4F3	FTM common control register H	FTCCONH		W/R	8	0x00
0xF4F4	FTM common start register L	FTCSTRL	FTCSTR	W	8/16	0x00
0xF4F5	FTM common start register H	FTCSTRH		W	8	0x00
0xF4F6	FTM common stop register L	FCSTPL	FCSTR	W	8/16	0x00
0xF4F7	FTM common stop register H	FCSTPH		W	8	0x00
0xF4F8	FTM common status register L	FTCSTATL	FTCSTA-	R	8	0x00
0xF4F9	FTM common status register H	FTCSTATH		R	8	0x00

[Note]
Word access is available for the SFRs(Specific Function Registers) that have the word symbol. Specify the even address to word-access the registers.

9.2.2 FTMn Cycle Register (FTnP: n = 0, 1, 2, 3)

Address: 0xF400, 0xF402（FT1P）, 0xF404（FT2P）, 0xF406（FT3P）
Access: R/W
Access size: 16 bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnP															
Byte symbol	FTnPH								FTnPL							
Bit symbol	FTnP15	FTnP14	FTnP13	FTnP12	FTnP11	FTnP10	FTnP9	FTnP8	FTnP7	FTnP6	FTnP5	FTnP4	FTnP3	FTnP2	FTnP1	FTnP0
Access type	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

FTnP is a specific function register (SFR) to set the cycle (clock count) of FTMn.
The configurable range is 0x0001 to 0xFFFF (clock count: 2 to 65536).
Set this register after setting the operation mode using FTnMD[1:0] bits of FTMn mode register.

Description of bits

- **FTnP15-0** (Bit 15 to 0)

Mode	FTnP15-0	Description
TIMER CAPTURE PWM1 PWM2	0x0001-0xFFFF	Set one cycle to FTnP setting value + 1 clocks.

[Note]
When 0x0000 is written in this register, 0x0001 is set. The read value is also becomes 0x0001.

9.2.3 FTMn Event Register A (FTnEA: n = 0, 1, 2, 3)

Address: 0xF410（FT0EA）, 0xF412（FT1EA）, 0xF414（FT2EA）, 0xF416（FT3EA）
Access: R/W
Access size: 16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnEA															
Byte symbol	FTnEAH								FTnEAL							
Bit symbol	FTnEA15	FTnEA14	FTnEA13	FTnEA12	FTnEA11	FTnEA10	FTnEA9	FTnEA8	FTnEA7	FTnEA6	FTnEA5	FTnEA4	FTnEA3	FTnEA2	FTnEA1	FTnEA0
Access type	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

FTnEA is a specific function register (SFR) to set the event timing of FTMn or display the capture data.
Set this register after setting the operation mode using FTnMD[1:0] bits of FTMn mode register. In the CAPTURE mode, this is a read-only register. It cannot be written.

Description of bits

- **FTnEA15-0** (Bit 15 to 0)

Mode	FTnEA15-0	Description
TIMER	0x0000-0xFFFF	Set a count value to generate an interrupt. (Interrupt timing is FTnEA setting value + 1) This value must be less than the cycle register FTnP.
CAPTURE	0x0000-0xFFFF	The captured count value is stored. When it is read, FTnFLGA bit of FTMn status register(FTnSTAT) and FTnISA bit of FTMn interrupt status register(FTnINTS) is cleared. In the CAPTURE mode, writing to FTnEA is invalid.
PWM1	0x0000-0xFFFF	Set the duty of the PWM signal driven from FTMnP pin. The duty in the PWM cycle becomes [the value set in this register +1]. Duty 100% is configurable.
PWM2	0x0000-0xFFFE	Set the duty of the PWM signal driven from FTMnP pin and FTMnN pin. The duty in the PWM cycle becomes [the value set in this register +1]. Set the value smaller than that set in the cycle register FTnP.

9.2.4 FTMn Event Register B (FTnEB: n = 0, 1, 2, 3)

Address: 0xF420（FT0EB）, 0xF422（FT1EB）, 0xF424（FT2EB）, 0xF426（FT3EB）
Access: R/W
Access size: 16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnEB															
Byte symbol	FTnEBH								FTnEBL							
Bit symbol	FTnEB15	FTnEB14	FTnEB13	FTnEB12	FTnEB11	FTnEB10	FTnEB9	FTnEB8	FTnEB7	FTnEB6	FTnEB5	FTnEB4	FTnEB3	FTnEB2	FTnEB1	FTnEB0
Access type	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

FTnEB is a specific function register (SFR) to set the event timing of FTMn or display the capture data.
Set this register after setting the operation mode using FTnMD[1:0] bits of FTMn mode register.
In the CAPTURE mode, this is a read-only register. It cannot be written.

Description of bits

- **FTnEB15-0** (Bit 15 to 0)

FTnMD	FTnEB15-0	Description
TIMER	0x0000-0xFFFF	Set a count value to generate an interrupt. (Interrupt timing is FTnEB setting value + 1) This value must be less than the cycle register FTnP.
CAPTURE	0x0000-0xFFFF	The captured count value is stored. When it is read, FTnFLGB bit of FTMn status register(FTnSTAT) and FTnISB bit of FTMn interrupt status register(FTnINTS) are cleared.In the CAPTURE mode, writing to this register FTnEB is invalid.
PWM1	0x0000-0xFFFF	Set the duty of the PWM signal driven from FTMnN pin. The duty in the PWM cycle becomes [the value set in this register +1]. Duty 100% is configurable.
PWM2	-	In this mode, set 0x0000 to FTnEB.

9.2.5 FTMn Dead Time Register (FTnDT: n = 0, 1, 2, 3)

Address: 0xF430（FT0DT）, 0xF432（FT1DT）, 0xF434（FT2DT）, 0xF436（FT3DT）
Access: R/W
Access size: 16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnDT															
Byte symbol	FTnDTH								FTnDTL							
Bit symbol	FTnDT15	FTnDT14	FTnDT13	FTnDT12	FTnDT11	FTnDT10	FTnDT9	FTnDT8	FTnDT7	FTnDT6	FTnDT5	FTnDT4	FTnDT3	FTnDT2	FTnDT1	FTnDT0
Access type	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

FTnDT is a specific function register (SFR) to set the dead time of timer output.
Set this register after setting the operation mode using FTnMD[1:0] bits of FTMn mode register.

Description of bits

- **FTnDT15-0** (Bit 15 to 0)

Mode	FTnDT15-0	Description
TIMER PWM1/2	0x0000-0xFFFF	Set the dead time of timer output or PWM output (timing of FTnDT setting value + 1). It is enabled when the FTnDTENP bit and FTnDTENN bit of FTMn mode register(FTnMOD) is "1."
CAPTURE	*	This register is disabled

9.2.6 FTMn Counter Register (FTnC: n = 0, 1, 2, 3)

Address: 0xF440 (FT0C), 0xF442 (FT1C), 0xF444 (FT2C), 0xF446 (FT3C)
Access: R/W
Access size: 16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnC															
Byte symbol	FTnCH								FTnCL							
Bit symbol	FTnC15	FTnC14	FTnC13	FTnC12	FTnC11	FTnC10	FTnC9	FTnC8	FTnC7	FTnC6	FTnC5	FTnC4	FTnC3	FTnC2	FTnC1	FTnC0
Access type	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

FTnC is a specific function register (SFR) to display the counter value of FTMn.
When writing to this register, the counter is cleared to ”0x0000”.

When writing to FTnC with word access or to FTnCL with byte access, FTnC is cleared to be "0x0000" regardless of the write data.

9.2.7 FTMn Status Register (FTnSTAT: n = 0, 1, 2, 3)

Address: 0xF450（FT0STAT）, 0xF452（FT1 STAT）, 0xF454（FT2 STAT）, 0xF456（FT3 STAT）
Access: R/W
Access size: 8 bit
Initial value: 0x0030

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FTnSTAT							
Bit symbol	-	-	-	-	-	-	-	-	FTnSTA	FTnFLGC	FTnFLGB	FTnFLGA	-	-	-	FTnUD
Access type	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	1	1	0	0	0	0

FTnSTAT is a specific function register (SFR) to indicate the state of FTMn.

Description of bits

- **FTnUD** (Bit 0)
  This is a bit to indicate the state of the completion after generating an update request of the cycle register or event register by writing "1" to FTCUDn bit of FTM common update register (FTCUD). When the transfer is completed, this bit is cleared automatically.

Mode	FTnUD	Description
TIMER CAPTURE PWM1/2	0	Update completed (initial value)
	1	Requesting update

- **FTnFLGA** (Bit 4)
  This is a bit to indicate the state of event timing A of FTMn.

Mode	FTnFLGA	Description
TIMER PWM1/2	0	Counter value < Value of event register A
	1	Counter value ≥ Value of event register A (initial value)
CAPTURE	0	Capture data not available
	1	Capture data available. When FTnEA is read, it is cleared.

- **FTnFLGB** (Bit 5)
  This is a bit to indicate the state of event timing B of FTMn.

Mode	FTnFLGB	Description
TIMER PWM1/2	0	Counter value < Value of event register B
	1	Counter value ≥ Value of event register B (initial value)
CAPTURE	0	Capture data not available
	1	Capture data available. When FTnEB is read, it is cleared.

- **FTnFLGC** (Bit 6)
  This is a bit to indicate whether the next event start is enable or disable while a counter selected by FTnCST bit of FTMn trigger register 0 is being stopped. This bit is cleared when reading the counter register FTnC.

Mode	FTnFLGC	Description
TIMER PWM1/2 CAPTURE	0	Start by event trigger enabled (initial value)
	1	Start by event trigger disabled

- **FTnSTA** (Bit 7)
  This is a bit to indicate the operation state of FTMn.

Mode	FTnSTA	Description
TIMER CAPTURE PWM1/2	0	Counter stopped (initial value)
	1	Counter running

9.2.8 FTMn Mode Register (FTnMOD: n = 0, 1, 2, 3)

Address: 0xF460(FT0MODL/FT0MOD), 0xF461(FT0MODH),
0xF462(FT1MODL/FT1MOD), 0xF463(FT1MODH),
0xF464(FT2MODL/FT2MOD), 0xF465(FT2MODH),
0xF466(FT3MODL/FT3MOD), 0xF467(FT3MODH)
Access: R/W
Access size: 8/16 bit
Initial value: 0x4000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTnMOD															
Byte symbol	FTnMODH								FTnMODL							
Bit symbol	FTnOSL1	FTnOSL0	FTnOSNN	FTnOSNP	-	-	-	FTnSTPO	FTnOST	-	FTnDTENN	FTnDTENP	-	-	FTnMD1	FTnMD0
Access type	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnMOD is a specific function register (SFR) to set the function of FTMn.

Description of bits

- **FTnMD1-0** (Bit 1 to 0)
  These are bits to set the mode of FTMn.

Mode	FTnMD1-0	Description
-	0	TIMER mode (initial value)
	1	CAPTURE mode
	2	PWM1 mode
	3	PWM2 mode

- **FTnDTENP** (Bit 4)
  This is a bit to enable the dead time of FTMnP.

Mode	FTnDTENP	Description
TIMER PWM1/2	0	Dead time disabled (initial value)
	1	Dead time enabled
CAPTURE	*	This bit is invalid

- **FTnDTENN** (Bit 5)
  This is a bit to enable the dead time of FTMnN.

Mode	FTnDTENN	Description
TIMER PWM1/2	0	Dead time disabled (initial value)
	1	Dead time enabled
CAPTURE	*	This bit is invalid

- **FTnOST** (Bit 7)
  This is a bit to set the repeat/one-shot mode of FTMn.

Mode	FTnOST	Description
TIMER PWM1/2	0	Repeat mode (initial value)
	1	One-shot mode
CAPTURE	0	Auto mode Even if the capture is performed once, the data of EA and EB is overwritten (updated) when the next capture is performed. When the counter goes round, it restarts from 0.
	1	Single mode Once captured into EA or EB, the next capture is not performed before read. When the counter goes round, it stops.

- **FTnSTPO** (Bit 8)
  This is a bit to set the output state of FTMnN pin and FTMnP pin while the FTMn is stopped.

Mode	FTnSTPO	Description
TIMER PWM1/2	0	When the FTnOSNP and FTnOSNN are "0", holds the output level "L" while the FTMn is stopped. When the FTnOSNP and FTnOSNN are "1", holds the output level "H" while the FTMn is stopped. If restarted the FTMn without clearing the counter, it holds the output level until the next cycle.
	1	Holds the current output level while the FTMn is stopped. When restarted the FTMn without clearing the counter, the output depends on the counter value. When the counter is cleared, the output depends on the setting of FTnOSNP and FTnOSNN.
CAPTURE	*	This bit is invalid

- **FTnOSNP** (Bit 12)**, FTnOSNN** (Bit 13)
  These are bits to reverse FTM output.
  They reverses the signal selected by FTnOSL0(bit14) and FTnOSL1(bit15).

FTnOSNP/ FTnOSNN	Description
0	Does not reverse the output. (Initial value)
1	Reverses the output.

- **FTnOSL0** (Bit 14)**, FTnOSL1** (Bit 15)
  These are bits to select phases of signal output from FTMnN pin and FTMnP pin.

FTnOSL1	FTnOSL0	Description	
		FTMnN pin output	FTMnP pin output
0	0	Output Negative phase from the both pins	
0	1	Output Negative phase (initial value)	Output Positive phase (initial value)
1	0	Output Positive phase	Output Negative phase
1	1	Output Positive phase from the both pins	

9.2.9 FTMn Clock Register (FTnCLK: n=0,1,2,3)

Address: 0xF470(FT0CLKL/FT0CLK), 0xF471(FT0CLKH),
0xF472(FT1CLKL/FT1CLK), 0xF473(FT1CLKH),
0xF474(FT2CLKL/FT2CLK), 0xF475(FT2CLKH),
0xF476(FT3CLKL/FT3CLK), 0xF477(FT3CLKH)
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 symbol	FTnCLK															
Byte symbol	FTnCLKH								FTnCLKL							
Bit symbol	-	-	-	-	-	FTnXCK2	FTnXCK1	FTnXCK0	-	FTnCKD2	FTnCKD1	FTnCKD0	FTnEX	-	-	FTnCK0
Access type	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnCLK is a specific function register (SFR) to set the function of FTMn.

Description of bits

- **FTnCK0** (Bit 0)
  This is a bit to select the timer clock source(FTnTCK) of FTMn. The timer clock source is used for sampling and edge detection for other external triggers.

Mode	FTnCK	Description
TIMER CAPTURE PWM1/2	0	LSCLK (initial value)
	1	HSCLK

- **FTnEX** (Bit 3)
  This is a bit to select the count clock(FTnCK) of FTMn.
  The count clock is used for counting operation and waveform output control.

Mode	FTnEX	Description
TIMER CAPTURE PWM1/2	0	Count the timer clock selected in the FTnCK0 bit with the dividing ratio selected in the FTnCKD2-0 bits. (initial value)
	1	Count with the rising edge of the external triggers (EXTRG0 to EXTRG7) selected in the FTnXCK2-0 bits.

- **FTnCKD2-0** (Bit 6 to 4)
  These are bits to select the dividing ratio of the count clock source of FTMn.
  When "1" is selected in the FTnEX bit, the settings of these bits are invalid.

Mode	FTnCKD	Description
TIMER CAPTURE PWM1/2	0	1 dividing (initial value)
	1	2 dividing
	2	4 dividing
	3	8 dividing
	4	16 dividing
	5	32 dividing
	6	64 dividing
	7	128 dividing

- **FTnXCK2-0** (Bit 10 to 8)
  These are bits to select the timer clock source when the external trigger is selected(FTnEX = 1).

Mode	FTnXCK	Description
TIMER CAPTURE PWM1/2	0	EXTRG0 (initial value)
	1	EXTRG1
	2	EXTRG2
	3	EXTRG3
	4	EXTRG4
	5	EXTRG5
	6	EXTRG6
	7	EXTRG7

9.2.10 FTMn Trigger Register 0 (FTnTRG0: n = 0, 1, 2, 3)

Address: 0xF480(FT0TRG0L/FT0TRG0), 0xF481(FT0TRG0H),
0xF482(FT1TRG0L/FT1TRG0), 0xF483(FT1TRG0H),
0xF484(FT2TRG0L/FT2TRG0), 0xF485(FT2TRG0H),
0xF486(FT3TRG0L/FT3TRG0), 0xF487(FT3TRG0H)
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 symbol	FTnTRG0															
Byte symbol	FTnTRG0H								FTnTRG0L							
Bit symbol	-	FTnEST1	FTnEST0	FTnSTSS	FTnSTS3	FTnSTS2	FTnSTS1	FTnSTS0	-	FTnDCLH	FTnCST	-	FTnSPC	FTnSP	FTnSTC	FTnST
Access type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	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

FTnTRG0 is a specific function register (SFR) to set the function of FTMn.

Description of bits

- **FTnST** (Bit 0)
  This is a bit to select whether to start the counter by a trigger event.

Mode	FTnST	Description
TIMER CAPTURE PWM1/2	0	Starting counter disabled (initial value)
	1	Starting counter enabled

- **FTnSTC** (Bit 1)
  This is a bit to select whether to clear the counter when a trigger event of counter start occurs (only when the edge is selected in the FTnTRM1-0 bits). The setting of this bit is valid regardless of the setting of the FTnST bit. If an update request of FTnP, FTnEA, FTnEB, and FTnDT by the FTCUDn bit of FTM common update regiser (FTCUD) is generated when the trigger event occurs, updating of FTnP, FTnEA, FTnEB, and FTnDT is also executed at the same time as counter clearing. It is not cleared at emergency stop regardless of the this bit setting. The trigger event is selected by FTnTRM1-0 bits. Set this FTnSTC bit to "0" when selecting "level" trigger event.

Mode	FTnSTC	Description
TIMER CAPTURE PWM1/2	0	Clearing counter disabled (initial value)
	1	Clearing counter enabled

- **FTnSP** (Bit 2)
  This is a bit to select whether to stop the counter by a trigger event.

Mode	FTnSP	Description
TIMER CAPTURE PWM1/2	0	Stopping counter disabled (initial value)
	1	Stopping counter enabled

- **FTnSPC** (Bit 3)
  This is a bit to select whether to clear the counter when a trigger event of counter stop occurs (only when the edge is selected in the FTnTRM1-0 bits). The setting of this bit is valid regardless of the setting of the FTnSP bit. If an update request of FTnP, FTnEA, FTnEB, and FTnDT by the FTCUDn bit of FTM common update regiser (FTCUD) is generated when the trigger event occurs, updating of FTnP, FTnEA, FTnEB, and FTnDT is executed at the same time as counter clearing. It is not cleared at emergency stop regardless of the this bit setting.

Mode	FTnSPC	Description
TIMER CAPTURE PWM1/2	0	Clearing counter disabled (initial value)
	1	Clearing counter enabled

- **FTnCST** (Bit 5)
  This is a bit to select the operation mode of counter start by a trigger event.

Mode	FTnCST	Description
TIMER CAPTURE PWM1/2	0	A trigger event always start the counter when it is stopped (except for emergency stop) (initial value)
	1	A trigger event does not start the counter before FTnC is read when it is stopped (except for emergency stop)

- **FTnDCLH** (Bit 6)
  This is a bit to disable counter clearing by a trigger event when the output pin FTMnP of FTMn is in the "H" level.

Mode	FTnDCLH	Description
TIMER PWM1/2	0	Clearing enabled regardless of the output level of the output pin FTMnP (initial value)
	1	Clearing disabled when the output level of the output pin FTMnP is "H"
CAPTURE	*	This bit is invalid

- **FTnSTSS, FTnSTS3-0** (Bit 12 to 8)
  These are bits to select the rigger event source of FTMn. Do not select the same source, for example, FTM0 in the setting of FTM0.

Mode	FTnSTS*					Description
	S	**3**	**2**	**1**	**0**	
TIMER CAPTURE PWM1/2	0	0	0	0	0	External pin EXTRG0 (initial value)
	0	0	0	0	1	External pin EXTRG1
	0	0	0	1	0	External pin EXTRG2
	0	0	0	1	1	External pin EXTRG3
	0	0	1	0	0	Reserved
	0	0	1	0	1	Reserved
	0	0	1	1	0	Reserved
	0	0	1	1	1	Reserved
	0	1	0	0	0	Comparator interrupt CMP0OUT
	0	1	0	0	1	Reserved
	0	1	0	x	x	Reserved
	0	1	1	0	0	For the clock mutual monitoring in the safety function
	0	1	1	x	x	Reserved
	1	0	0	0	0	16bit Timer 0 interrupt TMH0INT
	1	0	0	0	1	16bit Timer 1 interrupt TMH1INT
	1	0	0	1	0	16bit Timer 2 interrupt TMH2INT
	1	0	0	1	1	16bit Timer 3 interrupt TMH3INT
	1	0	1	0	0	Reserved
	1	0	1	0	1	Reserved
	1	0	1	1	0	Reserved
	1	0	1	1	1	Reserved
	1	1	0	0	0	Functional Timer 0 interrupt FTM0INT
	1	1	0	0	1	Functional Timer 1 interrupt FTM1INT
	1	1	0	1	0	Functional Timer 2 interrupt FTM2INT
	1	1	0	1	1	Functional Timer 3 interrupt FTM3INT
	others					Reserved

- **FTnEST1-0** (Bit 14 to 13)
  These are bits to select the emergency stop trigger source of FTMn. These bits are effective only when FTnEMGEN bit of FTM common control regiser (FTCCON) is 1.

Mode	FTnEST	Description
TIMER CAPTURE PWM1/2	0	EXTRG0 (initial value)
	1	EXTRG4
	2	CMP0OUT
	3	Reserved

[Note]
EXTRGn is a signal for trigger by an external pin. CMP0OUT is a signal for trigger of the comparator. The timer interrupt request (TMHnINT) is an interrupt request signal not depending on whether the interrupt of the interrupt enable register is set to be disabled/enabled. The functional timer trigger output (FTMnINT) is an exclusive signal for event triggers.

9.2.11 FTMn Trigger Register 1 (FTnTRG1: n = 0, 1, 2, 3)

Address: 0xF490(FT0TRG1L/FT0TRG1), 0xF491(FT0TRG1H),
0xF492(FT1TRG1L/FT1TRG1), 0xF493(FT1TRG1H),
0xF494(FT2TRG1L/FT2TRG1), 0xF495(FT2TRG1H),
0xF496(FT3TRG1L/FT3TRG1), 0xF497(FT3TRG1H)
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 symbol	FTnTRG1															
Byte symbol	FTnTRG1H								FTnTRG1L							
Bit symbol	-	-	-	-	-	FTnTRF2	FTnTRF1	FTnTRF0	-	-	-	FTnEMGES	-	FTnTRM2	FTnTRM1	FTnTRM0
Access type	R	R	R	R	R	R/W	R/W	R/W	R	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

FTnTRG1 is a specific function register (SFR) to set the function of FTMn.

Description of bits

- **FTnTRM1-0** (Bit 1 to 0)
  These are bits to select the edge or level of the trigger event of FTMn
  These are enabled only when EXTRG0-7 or CMP0OUT is selected for the event trigger source. Otherwise, it is fixed to the rising edge.

Mode	FTnTRM2-0			Description	
				Counter start	Counter stop
TIMER CAPTURE PWM1/2	0	0	0	Rising edge	Rising edge (initial value)
	0	0	1	Falling edge	Rising edge
	0	1	0	Rising edge	Falling edge
	0	1	1	Falling edge	Falling edge
	1	0	0	High level	Low level
	1	0	1	Low level	High level
	1	1	0	High level	Low level
	1	1	1	Low level	High level

- **FTnEMGES** (Bit 4)
  This is a bit to select the edge of the emergency stop trigger of FTMn.

Mode	FTnEMGES	Description
TIMER CAPTURE PWM1/2	0	Rising edge (initial value)
	1	Falling edge

- **FTnTRF2-0** (Bit 10 to 8)
  These are bits to set the noise removal width of the external trigger of FTMn, used to delay the start by the noise filter or trigger.
  These bits are enabled to use when EXTRG0-7, CMP0OUT or RC1KHz is selected for the event trigger source or the external clock is selected in the FTnEX bit of FTMn clock register(FTnCLK). In other case, this function is disabled.
  Also, in addition to the noise filter function set by this FTnTRF2-0 bits, another noise filter function can be set by the external interrupt mode register 0. See Chapter 18 “External Interrupt Function” for details abou the external interrupt.

Mode	FTnTRF2-0	Description
TIMER CAPTURE PWM1/2	0	Noise filter disabled (initial value)
	1	Removal of noise for 2 clocks of Timer Clock
	2	Removal of noise for 4 clocks of Timer Clock
	3	Removal of noise for 8 clocks of Timer Clock
	4	Removal of noise for 16 clocks (approx. 0.5 us) of Timer Clock
	5	Removal of noise for 32 clocks (approx. 1 us) of Timer Clock
	6	Removal of noise for 64 clocks (approx. 2 us) of Timer Clock
	7	Removal of noise for 128 clocks (approx. 4 us) of Timer Clock

[Note]
- EXTRG0-7 is a signal for trigger by an external pin. CMP0OUT is a signal for trigger of the comparator.
- When using the CMP0OUT as an emergency stop trigger, the filter function specified in FTnTRF2-0 is invalid.
- When the counter start condition is set to the level, if the levels of the start condition and external trigger are the same, the counting operation is continued (restart counting up from 0) even if the stop condition in the one-shot mode is satisfied.
- Enable the trigger event after setting the noise filter function. Othewise, the trigger may occur immediately after setting this register.

9.2.12 FTMn Interrupt Enable Register (FTnINTE: n = 0, 1, 2, 3)

Address: 0xF4A0(FT0INTEL/FT0INTE), 0xF4A1(FT0INTEH),
0xF4A2(FT1INTEL/FT1INTE), 0xF4A3(FT1INTEH),
0xF4A4(FT2INTEL/FT2INTE), 0xF4A5(FT2INTEH),
0xF4A6(FT3INTEL/FT3INTE), 0xF4A7(FT3INTEH)
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 symbol	FTnINTE															
Byte symbol	FTnINTEH								FTnINTEL							
Bit symbol	-	-	-	-	-	FTnIOB	FTnIOA	FTnIOP	-	-	-	FTnIETR	FTnIETS	FTnIEB	FTnIEA	FTnIEP
Access type	R	R	R	R	R	R/W	R/W	R/W	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

FTnINTE is a specific function register (SFR) to control the interrupt and trigger output of FTMn.
When each bit of FTnINTEL is set to "1", the interrupt is enabled and notified to the interrupt controller.
When each bit of FTnINTEH is set to "1", trigger output is enabled and notified to the DMA controller, A/D converter, and other channels of FTIMER.

Description of bits

- **FTnIEP** (Bit 0)
  This is a bit to set whether or not to enable the cyclic interrupt of FTMn

Mode	FTnIEP	Description
TIMER CAPTURE PWM1/2	0	Cyclic interrupt disabled (initial value)
	1	Cyclic interrupt enabled

- **FTnIEA** (Bit 1)
  This is a bit to set whether or not to enable the event timing A interrupt of FTMn.

Mode	FTnIEA	Description
TIMER PWM1/2	0	Event timing A interrupt disabled (initial value)
	1	Event timing A interrupt enabled
CAPTURE	0	Capture A interrupt disabled
	1	Capture A interrupt enabled

- **FTnIEB** (Bit 2)
  This is a bit to set whether or not to enable the event timing B interrupt of FTMn.

Mode	FTnIEB	Description
TIMER PWM1	0	Event timing B interrupt disabled (initial value)
	1	Event timing B interrupt enabled
PMW2	0	Always set "0"
	1	Prohibited
CAPTURE	0	Capture B interrupt disabled
	1	Capture B interrupt enabled

- **FTnIETS** (Bit 3)
  This is a bit to set whether or not to enable the trigger counter stop interrupt of FTMn.

Mode	FTnIETS	Description
TIMER CAPTURE PWM1/2	0	Trigger counter stop interrupt disabled (initial value)
	1	Trigger counter stop interrupt enabled

- **FTnIETR** (Bit 4)
  This is a bit to set whether or not to enable the trigger counter start interrupt of FTMn.

Mode	FTnIETR	Description
TIMER CAPTURE PWM1/2	0	Trigger counter start interrupt disabled (initial value)
	1	Trigger counter start interrupt enabled

- **FTnIOP** (Bit 8)
  This is a bit to enable or disable triggers (FTMnINT) to other peripherals when the FTMn cycle matches to the FTnC counter.

Mode	FTnIOP	Description
TIMER CAPTURE PWM1/2	0	Cyclic trigger disabled (initial value)
	1	Cyclic trigger enabled

- **FTnIOA** (Bit 9)
  This is a bit to output the event timing A request of FTMn as a trigger (FTMnINT) to other peripherals.

Mode	FTnIOA	Description
TIMER CAPTURE PWM1/2	0	Event timing A trigger disabled (initial value)
	1	Event timing A trigger enabled

- **FTnIOB** (Bit 10)
  This is a bit to output the event timing B request of FTMn as a trigger (FTMnINT) to other peripherals.

Mode	FTnIOB	Description
TIMER CAPTURE PWM1/2	0	Event timing B trigger disabled (initial value)
	1	Event timing B trigger enabled

9.2.13 FTMn Interrupt Status Register (FTnINTS: n = 0, 1, 2, 3)

Address: 0xF4B0(FT0INTSL/FT0INTS), 0xF4B1(FT0INTSH),
0xF4B2(FT1INTSL/FT1INTS), 0xF4B3(FT1INTSH),
0xF4B4(FT2INTSL/FT2INTS), 0xF4B5(FT3INTSH),
0xF4B6(FT3INTSL/FT3INTS), 0xF4B7(FT5INTSH)
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 symbol	FTnINTS															
Byte symbol	FTnINTSH								FTnINTSL							
Bit symbol	-	-	-	-	-	-	-	-	-	-	FTnISES	FTnISTR	FTnISTS	FTnISB	FTnISA	FTnISP
Access type	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

FTnINTS is a specific function register (SFR) to indicate the interrupt status of FTMn.
FTnINTS is a read-only register. Writing to it has no effect.

Description of bits

- **FTnISP** (Bit 0)
  This is a bit to indicate the state of the cyclic interrupt of FTMn.

Mode	FTnISP	Description
TIMER CAPTURE PWM1/2	0	Cyclic interrupt has not occurred (initial value)
	1	Cyclic interrupt has occurred This bit is cleared when writing 1 to FTnICP bit of FTMn interrupt clear register (FTnINTC)

- **FTnISA** (Bit 1)
  This is a bit to indicate the state of the event timing A interrupt of FTMn.
  It indicates that the captured data is stored to FTnEA in the CAPTURE mode.

Mode	FTnISA	Description
TIMER PWM1/2	0	Event timing A interrupt has not occurred (initial value)
	1	Event timing A interrupt has occurred This bit is cleared when writing 1 to FTnICA bit of FTMn interrupt clear register (FTnINTC)
CAPTURE	0	Capture A interrupt has not occurred
	1	Capture A interrupt has occurred This bit is cleared when writing 1 to FTnICA bit or reading the FTMn event register A(FTnEA)

- **FTnISB** (Bit 2)
  This is a bit to indicate the state of the event timing B interrupt of FTMn.

Mode	FTnISB	Description
TIMER PWM1/2	0	Event timing B interrupt has not occurred (initial value)
	1	Event timing B interrupt has occurred This bit is cleared when writing 1 to FTnIB
CAPTURE	0	Capture B interrupt has not occurred
	1	Capture B interrupt has occurred Indicates that the captured data is stored to the FTMn event register B(FTnEB). This bit is cleared when writing 1 to FTnICB bit of FTMn interrupt clear register (FTnINTC) or reading FTnEB

- **FTnISTS** (Bit 3)
  This is a bit to indicate the state of the trigger counter stop interrupt of FTMn.

Mode	FTnISTS	Description
TIMER CAPTURE PWM1/2	0	Trigger counter stop interrupt has not occurred (initial value)
	1	Trigger counter stop interrupt has occurred This bit is cleared when writing 1 to FTnICTS bit of FTMn interrupt clear register (FTnINTC)

- **FTnISTR** (Bit 4)
  This is a bit to indicate the state of the trigger counter start interrupt of FTMn.

Mode	FTnISTR	Description
TIMER CAPTURE PWM1/2	0	Trigger counter start interrupt has not occurred (initial value)
	1	Trigger counter start interrupt has occurred This bit is cleared when writing 1 to FTnICTR bit of FTMn interrupt clear register (FTnINTC)

- **FTnISES** (Bit 5)
  This is a bit to indicate the state of the emergency stop interrupt of FTMn.

Mode	FTnISES	Description
TIMER CAPTURE PWM1/2	0	Emergency stop interrupt has not occurred (initial value)
	1	Emergency stop interrupt has occurred This bit is cleared when writing 1 to FTnICES bit of FTMn interrupt clear register (FTnINTC)

9.2.14 FTMn Interrupt Clear Register (FTnINTC: n = 0, 1, 2, 3)

Address: 0xF4C0(FT0INTCL/FT0INTC), 0xF4C1(FT0INTCH),
0xF4C2(FT1INTCL/FT1INTC), 0xF4C3(FT1INTCH),
0xF4C4(FT2INTCL/FT2INTC), 0xF4C5(FT2INTCH),
0xF4C6(FT3INTCL/FT3INTC), 0xF4C7(FT3INTCH)
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 symbol	FTnINTC															
Byte symbol	FTnINTCH								FTnINTCL							
Bit symbol	FTnIR	-	-	-	-	-	-	-	-	-	FTnICES	FTnICTR	FTnICTS	FTnICB	FTnICA	FTnICP
Access type	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

FTnINTC is a specific function register (SFR) to clear the interrupt status of FTMn.
When writing 1 to this bit, the target interrupt status is cleared.
When reading it, 0x0000 is always read.

Description of bits

- **FTnICP** (Bit 0)
  This is a bit to clear the cyclic interrupt of FTMn.

- **FTnICA** (Bit 1)
  This is a bit to clear the event timing A interrupt of FTMn.

- **FTnICB** (Bit 2)
  This is a bit to clear the event timing B interrupt of FTMn.

- **FTnICTS** (Bit 3)
  This is a bit to clear the trigger counter stop interrupt of FTMn.

- **FTnICTR** (Bit 4)
  This is a bit to clear the trigger counter start interrupt of FTMn.

- **FTnICES** (Bit 5)
  This is a bit to clear the emergency stop interrupt of FTMn.

- **FTnIR** (Bit 15)
  An interrupt request bit of FTMn.
  Write ”1” before exiting the interrupt vector. When there is any unprocessed interrupt source, the interrupt request is issued again.

9.2.15 FTM Common Update Register (FTCUD)

Address: 0xF4F0
Access: W
Access size: 8/16 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTCUD															
Byte symbol	FTCUDH								FTCUDL							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	FTCUD3	FTCUD2	FTCUD1	FTCUD0
Access type	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

FTCUD is a specific function register (SFR) to update FTnP, FTnEA, FTnEB, and FTnDT of FTMn during operation. This is an SFR common to each channel. Bit n corresponds to channel n.

Description of bits

- **FTCUDn** (Bit 3 to 0)
  This is a write-only bit to update FTnP, FTnEA, FTnEB, and FTnDT of FTMn during operation. To update FTnP, FTnEA, and FTnEB during operation, write "1" in this bit after setting these registers. By writing "1", the setting value is transferred to the internal buffers of FTnP, FTnEA, and FTnEB at the same time as the end of the cycle.

Mode	FTCUDn	Description
TIMER CAPTURE PWM1/2	0	Update request has not occurred (initial value)
	1	Update request has occurred

9.2.16 FTM Common Control Register (FTCCON)

Address: 0xF4F2
Access: R/W
Access size: 8/16 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTCCON															
Byte symbol	FTCCONH								FTCCONL							
Bit symbol	-	-	-	-	FT3SDN	FT2SDN	FT1SDN	FT0SDN	-	-	-	-	FT3EMGEN	FT2EMGEN	FT1EMGEN	FT0EMGEN
Access type	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

FTCCON is a specific function register (SFR) to set the function of FTMn.
This is an SFR common to each channel. Bit n corresponds to channel n.

Description of bits

- **FTnEMGEN** (Bit 3 to 0)
  This is a bit to enable emergency stop of FTMn.

Mode	FTnEMGEN	Description
TIMER PWM1/2	0	Emergency stop disabled (initial value)
	1	Emergency stop enabled
CAPTURE	*	This bit is invalid

- **FTnSDN** (Bit 11 to 8)
  This is a bit to mask the output of FTMn to L.

Mode	FTnSDN	Description
TIMER PWM1/2	0	Release the output mask (initial value)
	1	Set the output mask (fix output to L).
CAPTURE	*	This bit is disabled

9.2.17 FTM Common Start Register (FTCSTR)

Address: 0F4F4H
Access: W
Access size: 8/16 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTCSTR															
Byte symbol	FTCSTRH								FTCSTRL							
Bit symbol	-	-	-	-	FT3ETG	FT2ETG	FT1ETG	FT0ETG	-	-	-	-	FT3STR	FT2STR	FT1STR	FT0STR
Access type	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

FTCSTR is a specific function register (SFR) to set the function of FTMn.
This is an SFR common to each channel. Bit n corresponds to channel n.

Description of bits

- **FTnSTR** (Bit 3 to 0)
  This is a bit to start counting of FTMn by software. When "1" is written in this bit, a start trigger by software is generated.
  In the initial state after power-on, counting is stopped.

Mode	FTnSTR	Description
TIMER CAPTURE PWM1/2	0	Counting by software is retained (initial value)
	1	Counting by software is started

- **FTnETG** (Bit 11 to 8)
  This is a bit to enable counting stop/start by a trigger event. Control by the FTM stop register (FTCSTP) to disable it.
  The setting of this bit has no effect on clearing by the trigger event. It is controlled by the bit to enable/disable clearing in FTMn trigger register 0.
  Trigger operation is disabled in the initial state after power-on.

Mode	FTnETG	Description
TIMER CAPTURE PWM1/2	0	Counting state according to trigger operation (counting stop/start) is maintained (initial value)
	1	Trigger operation (counting stop/start) enabled

[Note]
Set the FTCSTR register when the FTMn stops (when the FTnSTA bit of FTnSTAT register is “0”).

9.2.18 FTM Common Stop Register (FTCSTP)

Address: 0F4F6H
Access: W
Access size: 8/16 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FTCSTP															
Byte symbol	FTCSTPH								FTCSTPL							
Bit symbol	-	-	-	-	FT3DTG	FT2DTG	FT1DTG	FT0DTG	-	-	-	-	FT3STP	FT2STP	FT1STP	FT0STP
Access type	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

FTCSTP is a specific function register (SFR) to set the function of FTMn. This is an SFR common to each channel.

Description of bits

- **FTnSTP** (Bit 3 to 0)
  This is a bit to stop counting of FTMn by software. When "1" is written in this bit, a stop trigger by software is generated.
  In the initial state after power-on, counting is stopped.

Mode	FTnSTP	Description
TIMER CAPTURE PWM1/2	0	Counting state according to software is retained (initial value)
	1	Counting by software is stopped

- **FTnDTG** (Bit 11 to 8)
  This is a bit to disable counting start/stop by a trigger event to disable counting by the trigger event. Control by the FTM stop register (FTCSTR) to enable it.
  Trigger operation is disabled in the initial state after power-on.

Mode	FTnDTG	Description
TIMER CAPTURE PWM1/2	0	Counting state according to trigger operation (counting stop/start) is retained (initial value)
	1	Trigger operation (counting stop/start) disabled and counting by trigger operation stopped

[Note]
Set the FTCSTP register when the FTMn is running (when the FTnSTA bit of FTnSTAT register is “1”).

9.2.19 FTM Common Status Register (FTCSTAT)

Address: 0xF4F8
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 symbol	FTCSTAT															
Byte symbol	FTCSTATH								FTCSTATL							
Bit symbol	-	-	-	-	FT3TGEN-	FT2TGEN-	FT1TGEN-	FT0TGEN-	-	-	-	-	FT3RUN	FT2RUN	FT1RUN	FT0RUN
Access type	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

FTCSTAT is a specific function register (SFR) to indicate the state of FTMn. This is an SFR common to each channel.

Description of bits

- **FTnRUN** (Bit 0 to 3)
  This is a bit to indicate the counting state of FTMn.
  In the initial state after power-on, counting is stopped.

Mode	FTnRUN	Description
TIMER CAPTURE PWM1/2	0	Counting stopped (initial value)
	1	Counting in progress

- **FTnTGEN** (Bit 11 to 8)
  This is a bit to check the setting state of FTMn.
  Trigger operation is disabled in the initial state after power-on.

Mode	FTnTGEN	Description
TIMER CAPTURE PWM1/2	0	Trigger operation (counting stop/start) disabled (initial value)
	1	Trigger operation (counting stop/start) enabled

9.3 Description of Operation

Each channel functional timer has following four operation modes.

TIMER mode:
The level of timer output pin changes in syncronization with the start of timer counter and the overflow, generates a inverting pulse signal with the timer cycle. An interrupt can be generated when the counter overflows.

CAPTURE mode:
A count value of the timer counter when a selected event occurs on the risig edge of the trigger is stored in the FTMn event register A (FTnEA). A count value of the timer counter when a selected event occurs on the falling edge of the trigger is stored in the FTMn event register B (FTnEB).

PWM1 mode:
Two PWM waveforms can be generated using FTMn event register A (FTnEA) as the duty value of the output signal FTMnP and the FTMn event register B (FTnEB) as the duty value of the output signal FTMnN. They have the same cycle with an aligned start edge.

PWM2 mode:
Using FTMn event register A (FTnEA) as a duty value of the output signal FTMnP, a complementary PWM waveform with the output signal FTMnN operating exclusively can be generated. Also, the dead time can be set by the FTMn dead time register (FTMnDT).

9.3.1 Common Sequence

To operate FTM, follow the settings 1 to 6 below as needed. Then, start operation by setting FTM common start register (FTCSTR).
During operation, it is possible to check the hardware states such as interrupt status and to update the cycle/event setting.

1: Mode setting (FTnMOD)
Select the TIMER/CAPTURE/PWM mode using the mode register (FTnMOD). Set also the repeating mode such as repeat/one-shot mode.

2: Clock setting (FTnCLK)
Select the timer clock and the counter clock. If the internal clock is selected for the counter clock, the dividing ratio can be also set.

3: Trigger setting (FTnTRG0/1)
Use this setting when starting/stopping the counter by event trigger. Select the event trigger source and the action for FTnTRG0 and the event trigger/emergency stop edge for FTnTRG1.

4: Interrupt setting (FTnINTE)
Set the interrupt source. Select from cycle/event (counter coincide, duty, capture) and trigger start/stop interrupt.

5: Cycle/event setting (FTnP, FTnEA, FTnEB, FTnDT)
Set the cycle, data for counter coincide, duty, and dead time.

	TIMER	CAPTURE	PWM1	PWM2
FTnP	Repeat period or timeout of one-shot			
FTnEA	Coincident interrupt setting value	(Capture data)	FTMnP duty	Duty
FTnEB	Coincident interrupt setting value		FTMnN duty	(Unused)
FTnDT	Dead time for output	(Unused)	Dead time for output	Dead time for output

The cycle is calculated as follows:

$$T_{priod} = \frac{FTnP + 1}{FTnCK\ [Hz]} \quad (FTnP: 0x0001 \text{ to } 0xFFFF)$$

6: Control start/stop

Allow the software start or event trigger reception. Also, set the emergency stop enable.
The counter operates at the rising edge of FTnCK. The software start/stop are synchronized by FTnCK. FTnSTAT is set to H after FTnCK1 cycle at start, and the counter starts operating after two cycles. At stop, the counter is stopped in FTnCK1 cycle, and FTnSTAT is set to L. The counter value is kept at this time. If started again, it restarts after one cycle. To clear the counter, use write access to FTnC.

7: Processing during operation (FTnSTAT/FTCSTAT/FTnINTS, FTMUD)

The state during operation can be checked in FTMn status register(FTnSTAT), FTM common status register(FTCSTAT) and FTMn interrupt status register(FTnINTS). To change the waveform of PWM, etc., set the cycle/event and applicable bit of FTM common update register(FTCUD). Then, it is updated in the next cycle. When FTnSDN bit of FTM common control register(FTCCON) is set, the output is masked to "L" forcibly.

9.3.2 Counter Operation

The counter in FTM performs operation common to each mode.
The counter is counted up to the setting value of the FTMn cycle register (FTnP).
When the mode is the repeat mode (the FTnOST bit of the FTMn mode register (FTnMOD) is "0") at the time of overflow, the counter is cleared and the counting operation is continued again. At overflow in one-shot mode (FTnOST bit of FTnMOD is "1"), the counter is cleared and stops counting.
The software or trigger event can start/stop counting.

9.3.2.1 Starting/Stopping Counting by Software

When writing "1" to the FTnSTR bit of the FTCSTR register, the FTnSTA bit of the FTnSTAT register showing the count status becomes "1", and the counting operation is started.
In the one-shot mode (the FTnOST bit of the FTnMOD register is "1"), the counting operation is stopped by overflow. The FTnSTA bit of the FTnSTAT register showing the count status automatically becomes "0".

When writing "1" to the FTnSTP bit of the FTCSTP register while the counter operation is in progress (the FTnSTA bit of the FTnSTAT register showing the count status is "1"), the counter stops its operation. To confirm the stop of the counter, check by the software that the FTnSTA bit of the FTnSTAT register is reset to "0". The counter value is maintained while the counter is not working.
After the counter is stopped, if "1" is written to the FTnSTR bit of the FTCSTR register again, it continues counting from the value at the time it stopped.
To clear the counter, execute writing to the FTnC register while it is stopped.
If subsequently restarting the counter, confirm that the FTnC register is reset to "0x0000", then write "1" to the FTnSTR bit of the FTCSTR register.

Update timing of the relevant registers:
If writing the registers when the timer stops and the counter is "0", they are updated at the timer start.
If writing the registers while the timer is running, they are updated in the next cycle of that the update is requested by FTCUDn bit of FTCUD register.
If writing the registers when the timer stops and the counter is not "0", the registers are not updated until the update is requested by FTCUDn bit. Update the registers by one of following two ways.

- Write the relevant registers after clearing the counter by setting the FTnCL register.
- Request updating the relevant registers by setting the FTCUDn bit of FTCUD register.

9.3.2.2 Starting/Stopping Counting by Trigger Event

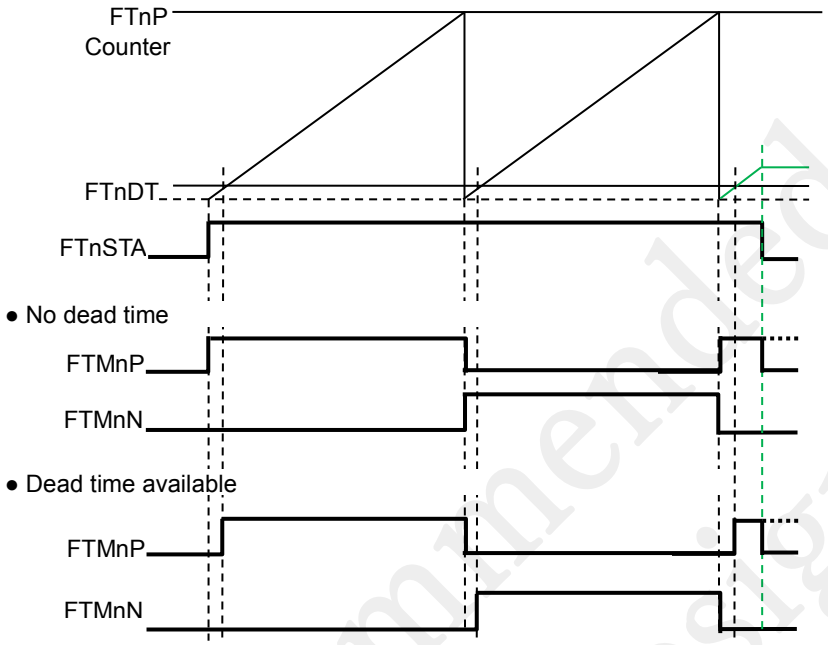
When the FTnETG bit of the FTM common start register (FTCSTR) is set to "1", the counter operation can be controlled by triggers.
Select a trigger by setting FTMn trigger register 0 and 1 (FTnTRG0, FTnTRG1).
The source of the trigger event can be selected from external interrupt, comparator output, timer interrupt, and other FTM triggers.
Depending on the selected trigger event, counter start, counter stop, counter start/stop and counter clearing can be selected.

9.3.3 TIMER Mode Operation

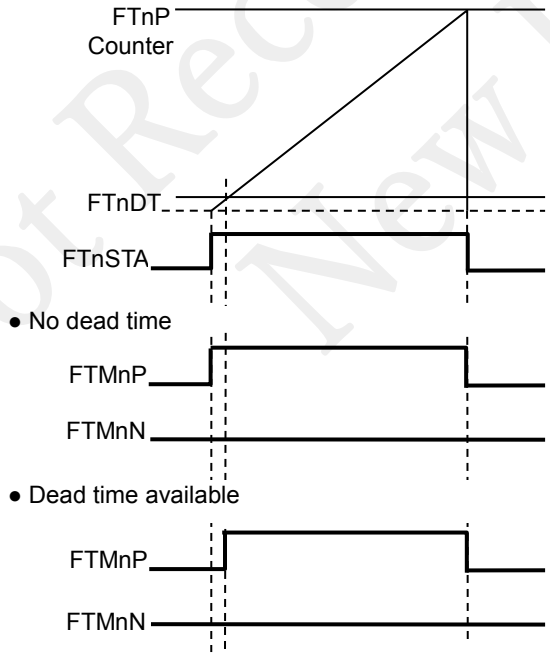
The TIMER mode controls the interrupt generation and output signal using the counter overflow.

9.3.3.1 Output Waveform in TIMER Mode

In the timer output repeat mode, the output is toggled for each cycle.
When the FTnRUN bit of the FTM common start register (FTCSTR) is set to "1" with the counter "0x0000", FTMnP starts operation from H and FTMnN at “L” level.
In the one-shot mode, it stops after outputting H pulse of one cycle from FTMnP. FTMnN is fixed to “L” level.
When FTnDTEN of the FTMn mode register (FTnMOD) is set to "1", output becomes “L” level until the count set in the FTMn dead time register (FTnDT) is exceeded after start of counting. The output waveforms with the setting of output reverse and phase.

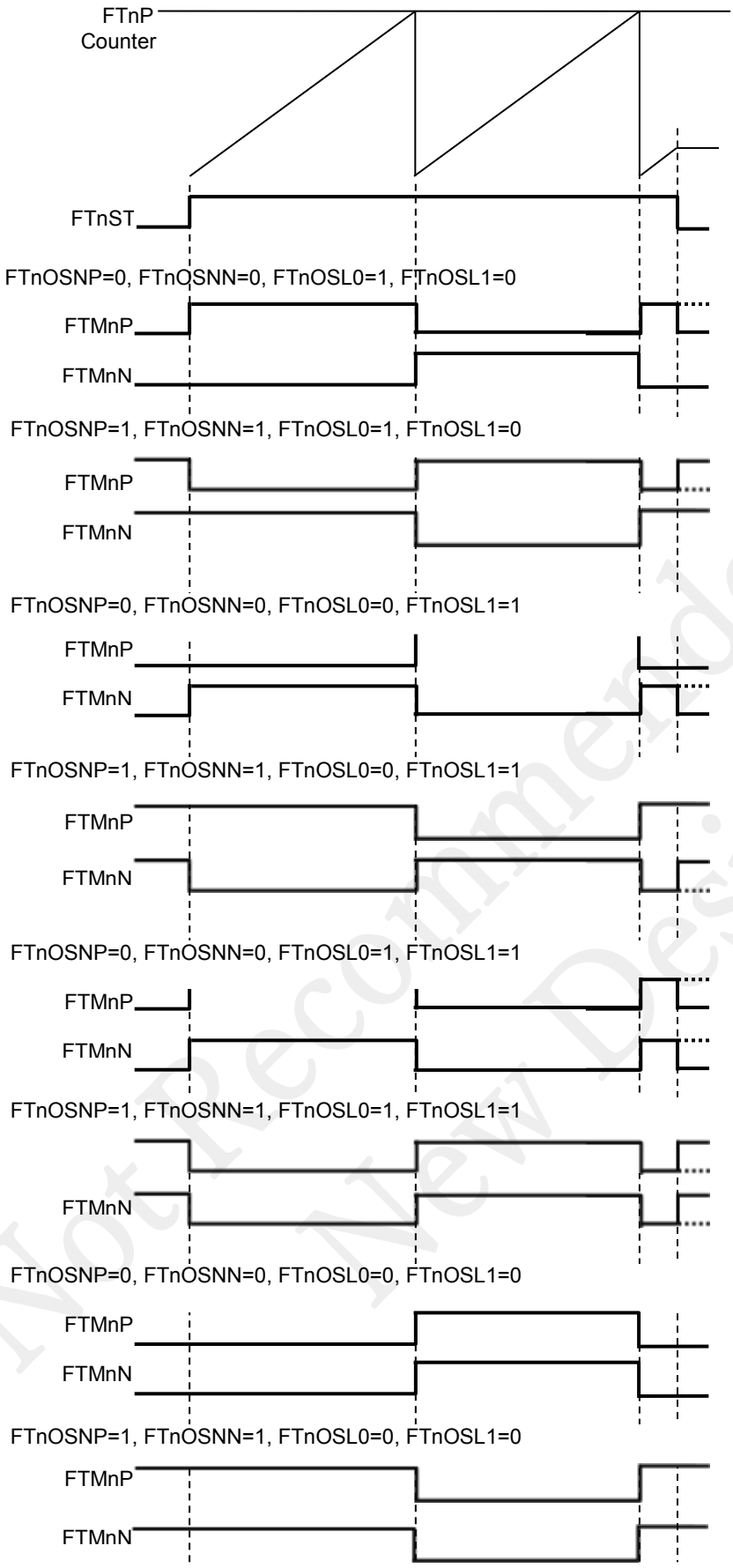


(a) TIMER mode output waveform (Repeat mode), The setting of output reverse and phase are initial



(b) TIMER mode output waveform (one-shot mode), The setting of output reverse and phase are initial

Figure 9-2 Output Waveform in Timer Mode(1/2)



(c) TIMER mode output waveform (Repeat mode), The setting of output reverse and phase are specifiied

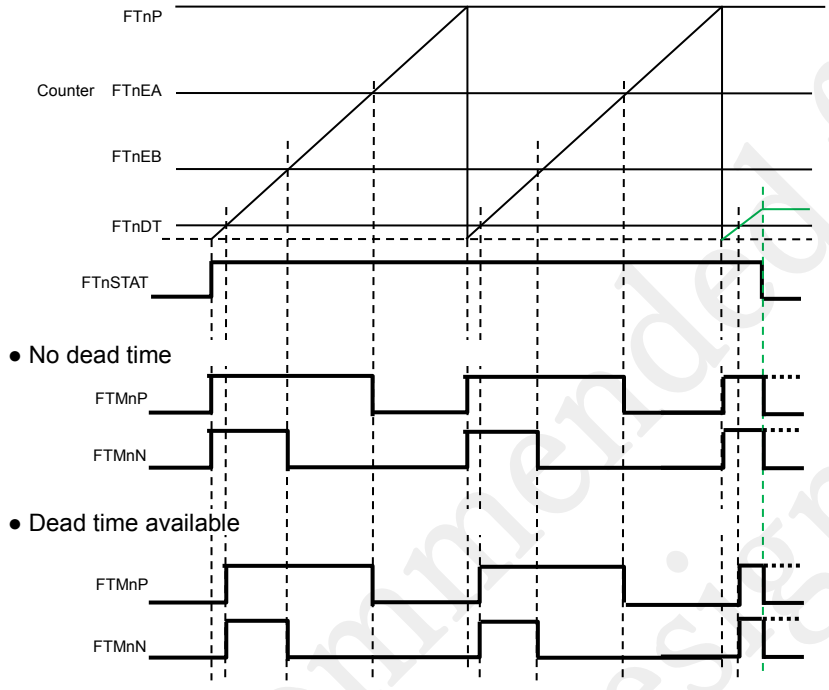
Figure 9-3 Output Waveform in Timer Mode(2/2)

9.3.4 PWM1 Mode Operation

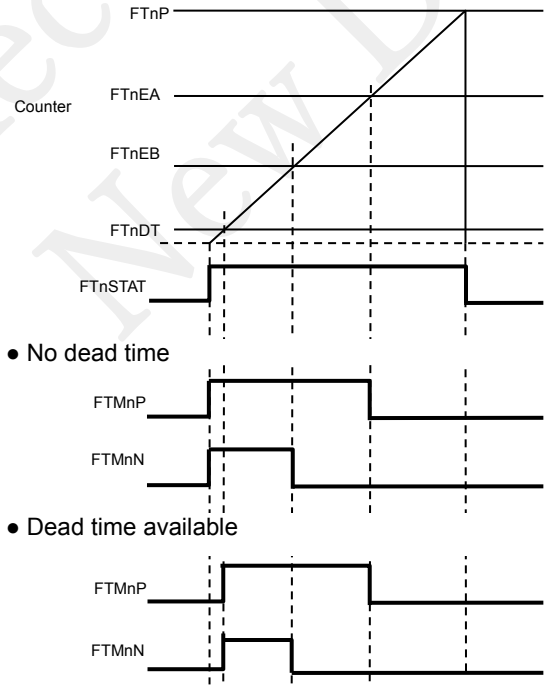
The PWM1 mode generates a synchronization output pulse with the cycle set in FTMn cycle register(FTnP). The duties of the output FTMnP and FTMnN are set in FTMn event register A(FTnEA) and FTMn event register B(FTnEB) respectively.

9.3.4.1 Output Waveform in PWM1 Mode

In the repeat mode, the initial values of both FTMnP and FTMnN are "L" level and change to "H" level at the time of start. Each of them changes to "L" level at the duty value. It changes to "H" level in the next cycle. This is repeated until they stop. In the one-shot mode, they automatically stop and change to "L" level after one cycle.
If the dead time is enabled, "L" level is output during the dead time from start of the counter.



(a) PWM1 mode output waveform (Repeat mode)



(b) PWM1 mode output waveform (one-shot mode)

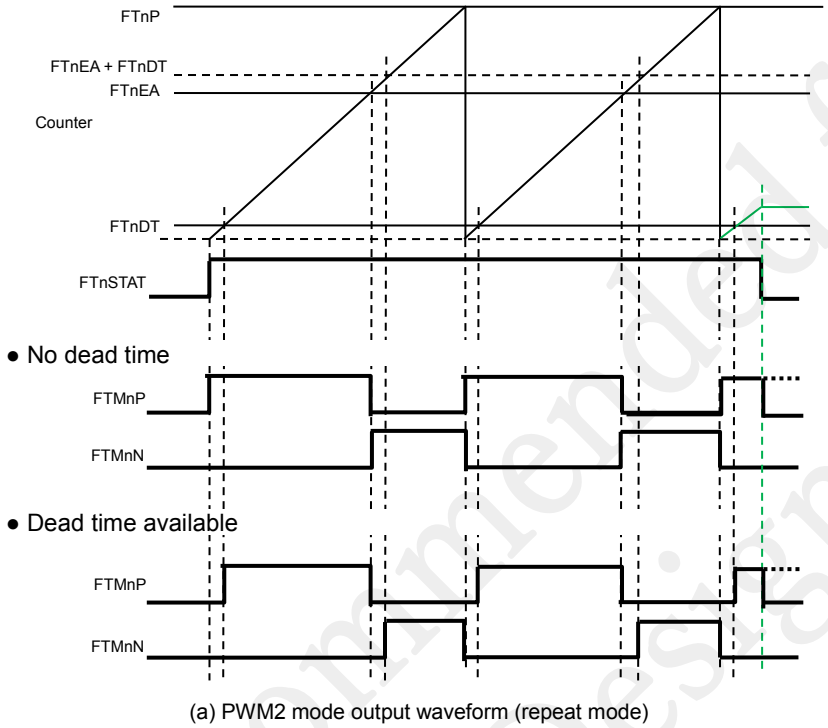
Figure 9-4 Output Waveform in PWM1 Mode

9.3.5 PWM2 Mode Operation

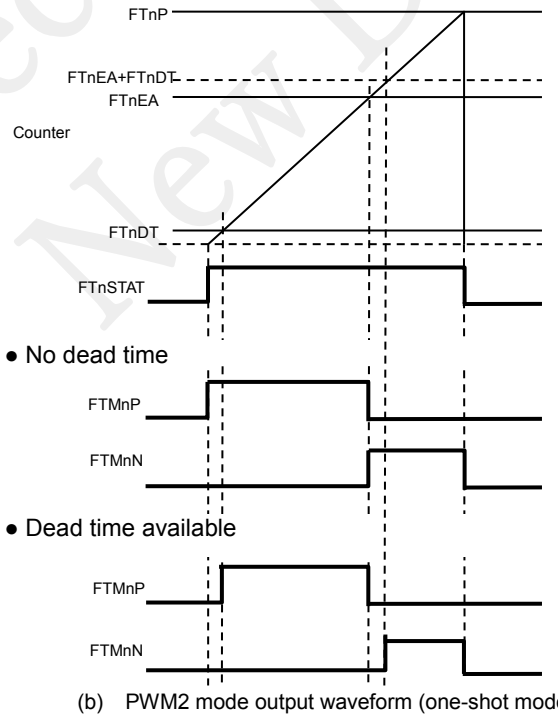
The PWM2 mode generates a complementary output pulse with the cycle set in FTMn cycle register(FTnP). The duties of output FTMnP/FTMnN are set in FTMn event register A(FTnEA). FTMn event register B(FTnEB) is not used.

9.3.5.1 Output Waveform in PWM2 Mode

In the repeat mode, the initial values of both FTMnP/FTMnN are L and the value of FTMnP becomes H at the time of start. FTMnP changes to L in the duty value and FTMnN to H. FTMnP changes to H and FTMnN to L again in the next cycle. This is repeated until they stop. In the one-shot mode, they automatically stop and change to L after one cycle. If the dead time is enabled, L is output during the dead time, from start of the counter in the case of FTMnP and from duty coincident in the case of FTMnN.



(a) PWM2 mode output waveform (repeat mode)



(b) PWM2 mode output waveform (one-shot mode)

Figure 9-5 Output Waveform in PWM2 Mode

9.3.6 CAPTURE Mode Operation

The CAPTURE mode stores the count value at the time when an event trigger source is generated, to the FTMn event register A(FTnEA) or FTMn event register B(FTnEB). The event trigger source to be captured is common to that used at counter start/stop.

Stored data in FTnEA	Counter value at the time when an event trigger rising edge is generated
Stored data in FTnEB	Counter value at the time when an event trigger falling edge is generated

9.3.6.1 Measurement Example in the CAPTURE Mode

The following example shows the measurement of the cycle and duty of PWM signal input from P02/EXTRG0 pin using CAPTURE mode and counter start/stop by trigger events.

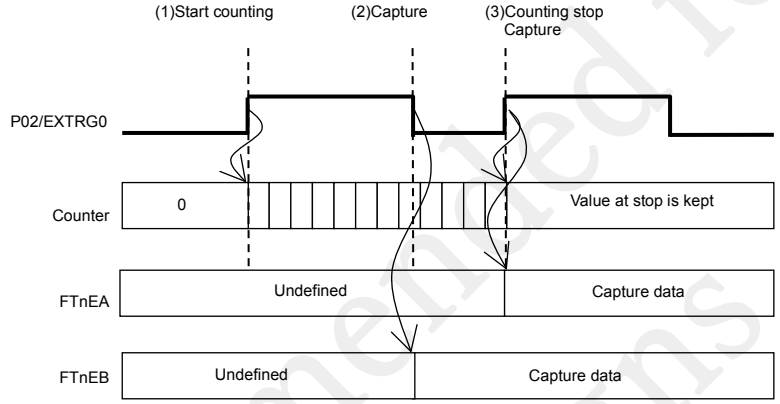


Figure 9-6 Measurement Example in the CAPTURE Mode

[Example for register settings]
- Select the CAPTURE mode by setting FTnMD[1:0] of FTMn mode register(FTnMOD) to “01b”
- To use an interrupt, enable the trigger counter stop interrupt by setting FTnIET bit of FTMn interrupt enable register(FTnINTE).
- Set FTnSTSS bit of FTMn trigger register 0(FTnTRG0) to “0”and FTnSTS[3:0] to ”0000” for selecting EXTRG0 as the trigger even source, also enable counter start function by setting FTnST bit to “1” and enable counter stop function by setting FTnSP bit to “1”.
- Select rising edge trigger for both counter start and stop by setting FTnTRM[2:0] of FTMn trigger register 1(FTnTRG1) to “00b”.
- Enable the capture trigger operation by setting FTnETG bit of FTM common start register(FTCSTR) to “1”.

[Example for hardware operation]
- The counter starts at rising edge of the signal input from EXTRG0 pin (1).
- The data of counter register(FTnC) is stored to the event register B(FTnEB) at a fulling edge of the signal (2).
- When the rising edge of EXTRG0 is detected again, the counter stops and an interrupt occurs. At this time, the data of counter is stored to the event register A(FTnEA) (3). In this example, the data of counter register(FTnC) corresponds to the cycle of PWM signal input from EXTRG0 pin and the data of FTnEB corresponds to the duty.

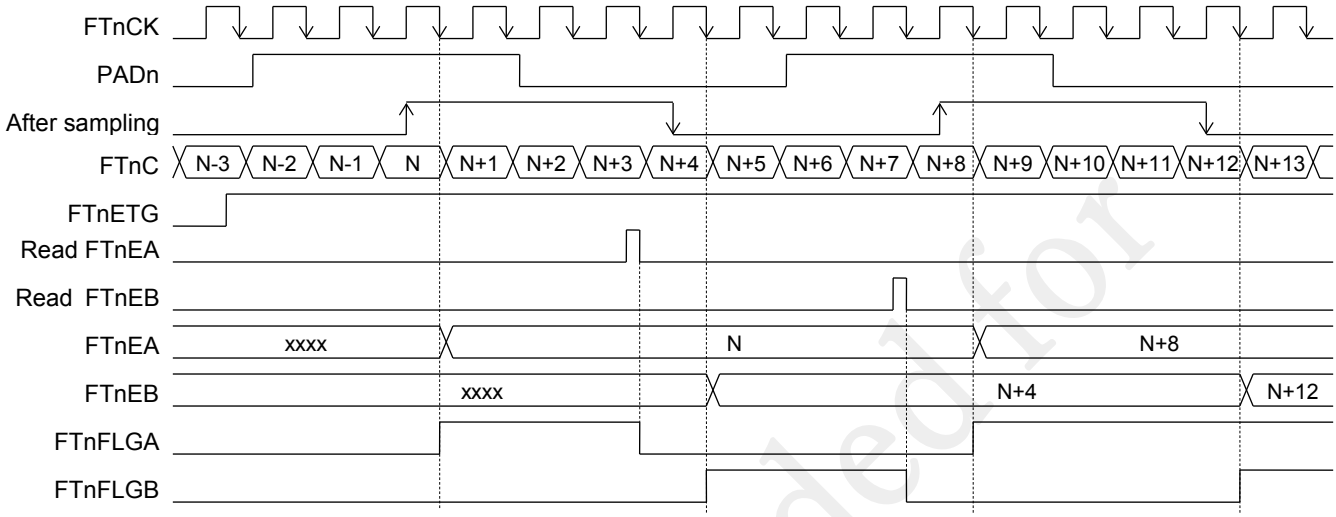
Also, the behaivor after the aforementioned capture operation dpends on the setting of FTnOST bit of FTnMOD register.

● In auto mode(FTnOST bit =0)

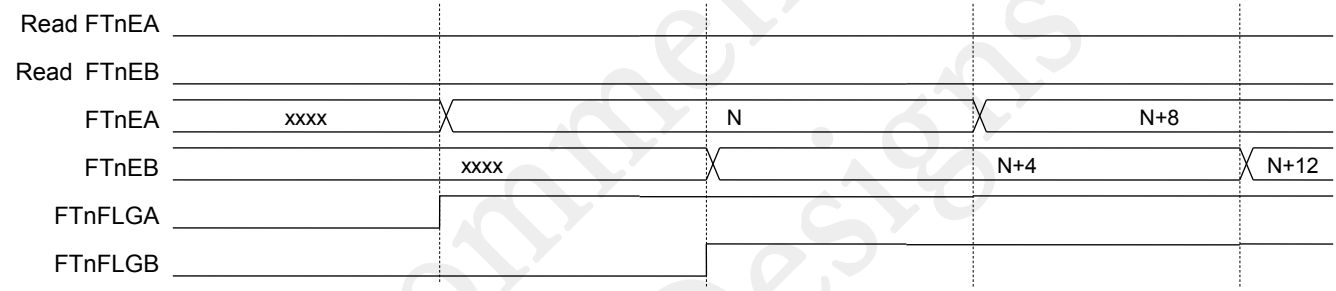
Data of FTnEA gets updated when the counter starts at the next rising edge.

●In single mode(FTnOST bit =1)

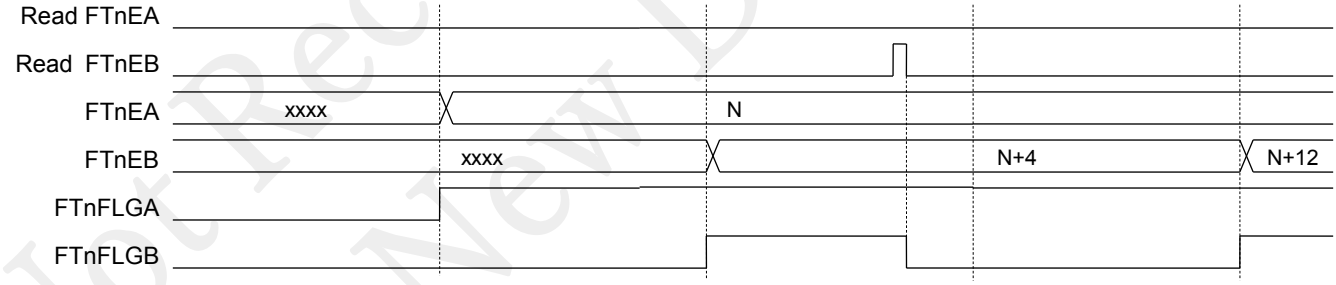
Data of FTnEA does not get updated when the counter starts at the next rising edge.



(a) When read before the next trigger(common to FTnOST = 0,1)



(b) When not read before the next trigger (FTnOST = 0)



(c) When not read before the next trigger (FTnOST = 1)

Figure 9-7 Operation Timing in CAPTURE Mode

9.3.7 Event/Emergency Stop Trigger Control

9.3.7.1 Trigger Signal

The Functional Timer(FTMn) can accept two types of trigger signals: event trigger and emergency stop trigger.
The event trigger is used as counter start/stop or trigger of capture. The trigger source can be selected from EXTRG0-7 (external pin), CMP0OUT (comparator output), TIMER0-7 interrupt or FTM0-3 trigger.
The emergency stop trigger is used to stop the timer operation. It stops the counter and sets output FTMnP/FTMnN to L. The trigger source can be selected from EXTRG0, EXTRG4, CMP0OUT.

To EXTRG0-7, output of the sampling controller of the port controller is connected. Sampling can be selected using the port controller register. FTMnF has also the sampling controller and it can be set in the FTnTRF2-0 bit of FTMn trigger register 1.
To CMP0OUT, output of the sampling controller of the comparator is connected. Sampling can be selected using the comparator register.
The source of interrupts of theTIMER0-7 and FTM0-3 can be selected using registers of each timer.

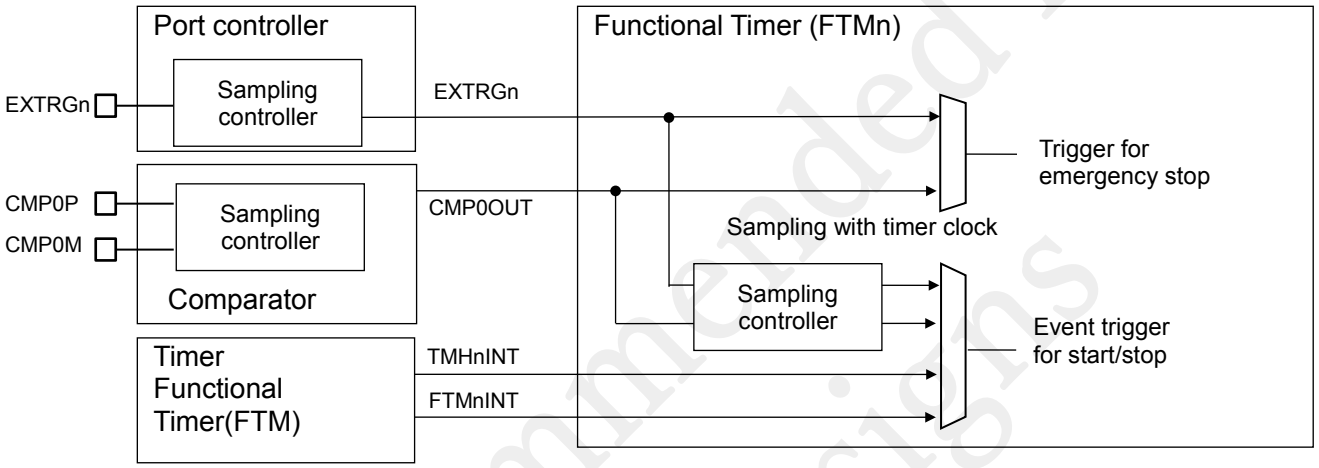


Figure 9-8 Input Route of Trigger Signal

9.3.7.2 Start/Stop Operations by Event Trigger

Here is the setting used to control the counter by event triggers.

1) FTMn trigger regiseter 0(FTnTRG0) setting
   Enable/Disable counter start/stop by event triggers
   Set whether or not to clear the counter at the time of start/stop by event triggers
   Set whether or not to accept the next counter start after stop by an event trigger
   Set whether or not to accept counter clearing when the output pin FTnP is H at the time of counter clearing by event triggers
   Set the event trigger source (EXTRG0-7, TIMER0-7INT, FTM0-3INT)
2) FTMn trigger regiseter 1(FTnTRG1) setting
   Set the edge/level of the event trigger for counter start
   Set the edge/level of the event trigger for counter stop
3) FTMn common start regiseter (FTCSTR) control
   Set FTnETG to ”1” to enter the waiting state for event triggers. (When level setting is applied to trigger start and the level is applicable, the counter operation is started as soon as "1" is set in FTnTGEN.)
   When "1" is set in FTnETG(trigger operation is enabled) with FTnSTR set to "1", counter is started by software.
   When the FTnSTP bit of the FTM common stop register is set to "0" during counter operation, the counter is stopped by software.

Set to the noise removal with set in FTnTRF2-0 in FTMn trigger register 1 or higher. Even when the noise filter is disabled in FTnTRF2-0, pulse of 1 clock or below may be sometimes removed, sometimes not.

Figure 9-8 shows the sampling timing of the external input.

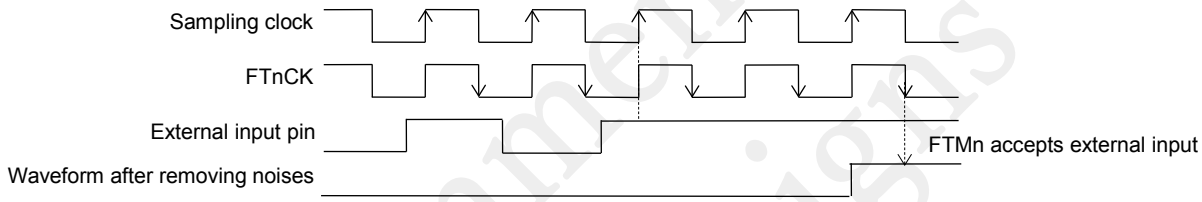
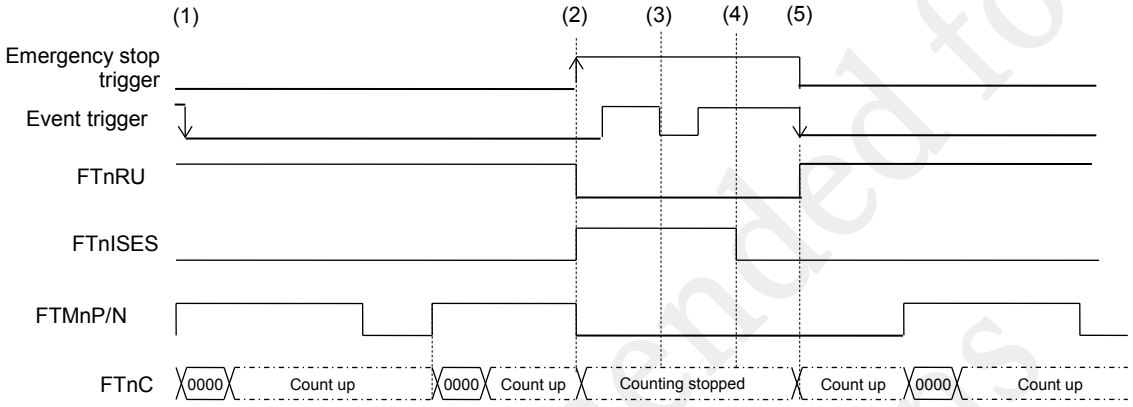


Figure 9-9 Sampling Timing of External Input

9.3.7.3 Emergency Stop Operation

When FTnEMGEN bit of FTM common control register(FTCCON) is set to "1", the emergency stop function is enabled. Set this bit after the trigger source is selected by FTnEST bit of FTMn trigger register 0(FTnTRG0).
If an emergency stop trigger input (rising edge) is detected, the counter stops, the output is set to L, and an emergency stop interrupt occurs.
To restart the counter operation, clear the emergency stop interrupt status by writing FTnICES bit of FTMn interrupt clear register to “1”.
Figure 9-9 shows the operation timing at emergency stop.
After emergency stop occurs, the counter is stopped after 1 timer clock and FTnISES bit of FTMn interrupt status register is set to “1”. When the FTnISES bit is 1, even if the event trigger of counter start occurs, it is not accepted. If the event trigger of counter start occurs after the FTnISES bit is cleared, counting up is restarted. To restart the counting operation by software, confirm that the FTnISES bit becomes "0".



(1) The counter operation starts at an event trigger (falling edge).
(2) The counter stops at an emergency stop trigger (rising edge). An emergency stop interrupt occurs.
(3) The event trigger is disabled due to the emergency stop in progress.
(4) Clear the emergency stop interrupt to enable the operation.
(5) The counter operation restarts at an event trigger (falling edge).
(The counter is not cleared in this example, so pulse output is restarted after one cycle)

Figure 9-10 Operation Timing Diagram at Emergency Stop

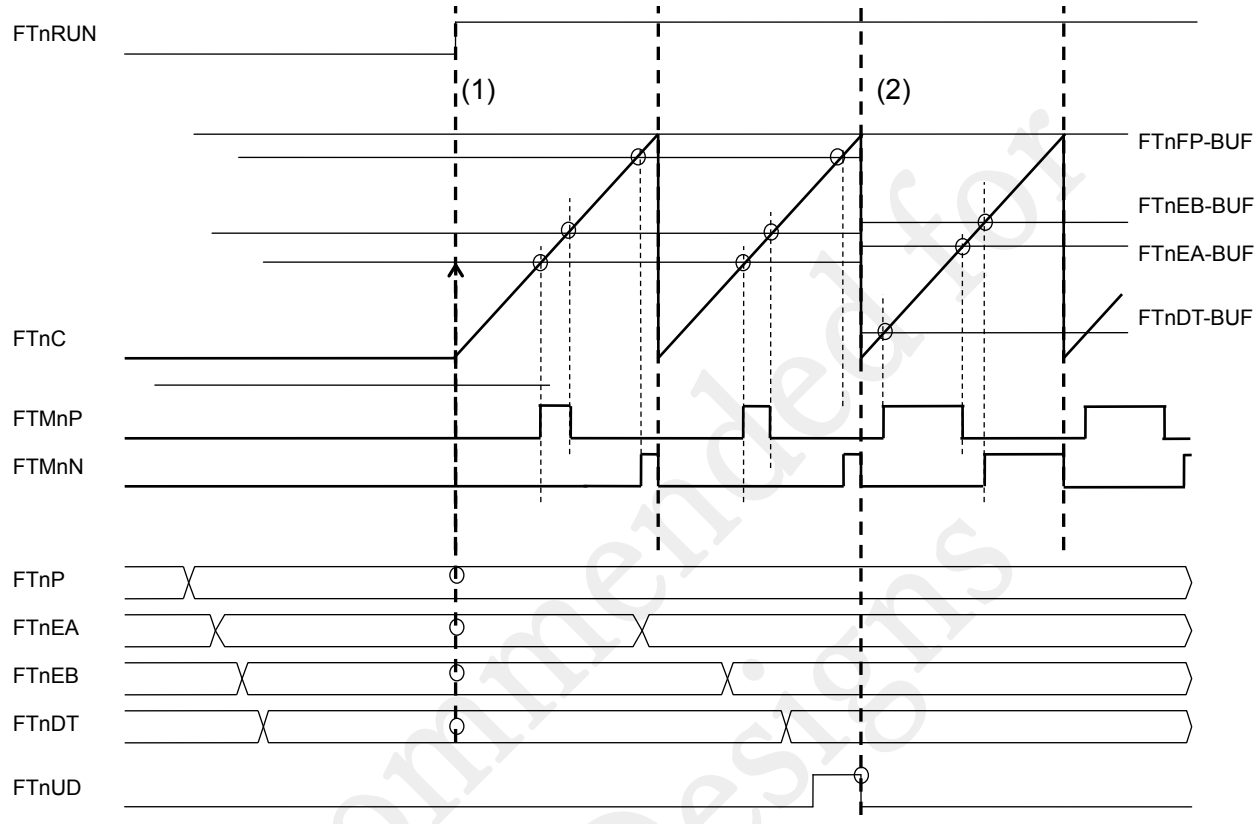
9.3.8 Output at Counter Stop

The states of FTMnP and FTMnN when the counter is stopped by software/event trigger depend on the setting of FTnSTPO bit of FTMn mode register(FTnMOD). If FTnSTPO is "0", FTMnP/FTMnN become "L" as soon as the counter stops. If the counter is restarted in this state, the FTMnP/FTMnN outputs keep ”L” during that cycle, and they change according to the counter value from the next cycle. When FTnSTPO bit is "1", FTMnP/FTMnN retain the states at the time of stop. When counting is restarted, the states change according to the counter value.
When the FTnSTC bit of FTMn trigger register(FTnTRG0) is set to "1" or the software clears the counter after counter stop, the counter value is counted up from ”0000”, and the output depends on the counter value.

9.3.9 Changing Cycle, Event A/B, and Dead Time during Operation

The cycle, event A/B, and dead time can be changed in the next cycle when the timer is counting.
To change them, set the registers to be changed (FTnP, FTnEA, FTnEB, FTnDT, etc.), then write "1" to the applicable bit in FTM common update register(FTCUD) to request updating. The values of the cycle, event A/B, buffer for dead time are updated at the beginning of the next cycle and the FTnUD bit of FTM status register(FTnSTAT) becomes "0".

Here is an example in the PWM2 mode (DTEN=1).



(1) The values set during stop update each buffer when the counter operation is started.
(2) During operation, each buffer is updated at the beginning of the next cycle for which PFUD is set to "1"

Figure 9-11 Updating Timing during Operation

9.3.10 Interrupt Source

This section describes the interrupt source and how to clear it.
When a interrupt enable bit(FTnIE*) of FTMn interrupt enable register(FTnINTE) is set to “1”, the interrupt is enabled. Note that the emergency stop interrupt enable does not exist. When the emergency stop is enabled, its interrupt is also enabled.
If the interrupt status is set to “1” for a source, clear each interrupt status bit(FTnIS*) by writing “1” to the each interrupt stuat clear bit(FTnIC*).
When the interrupt vector is used, clear each interrupt stauts bit(FTnIS*) at the end of the interrupt processing (when exiting the interrupt vector).

Name	Mode	Status	How to clear
Period coincident interrupt	ALL	FTnISP	Write ”1” to FTnICP
Event A coincident interrupt	TIMER/PWM1/PWM2	FTnISA	Write ”1” to FTnICA
Capture A interrupt	CAPTURE	FTnISA	Write ”1” to FTnICA or read FTnEA
Event B coincident interrupt	TIMER/PWM1	FTnISB	Write ”1” to FTnICB
Capture B interrupt	CAPTURE	FTnISB	Write ”1” to FTnICB or read FTnEB
Trigger stop interrupt	ALL	FTnISTS	Write ”1” to FTnICTS
Trigger start interrupt	ALL	FTnISTR	Write ”1” to FTnICTR
Emergency stop interrupt	ALL	FTnISES	Write ”1” to FTnICES

The interrupt trigger output can be selected from cycle coincident interrupt/Event A coincident interrupt/Event B coincident interrupt.

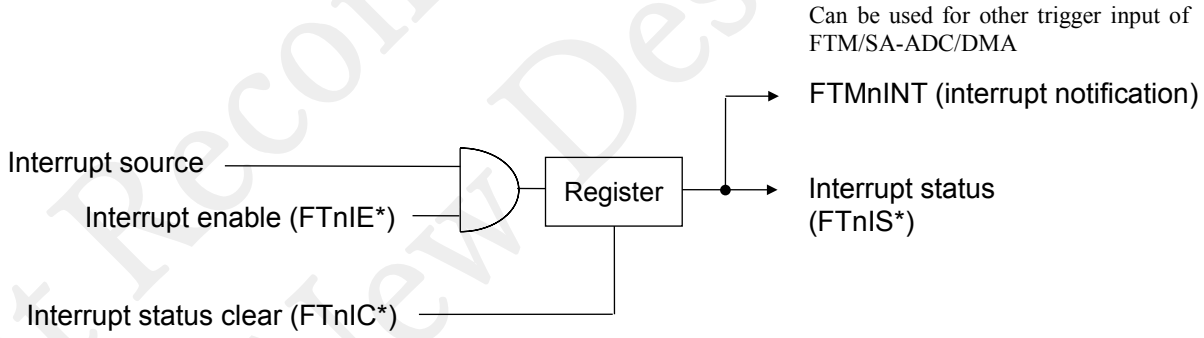


Figure 9-12 Interrupt Control Signal

Chapter 10 Watchdog Timer

10. Watchdog Timer

10.1 General Description

ML62Q1000 series has the watchdog timer (WDT) to detect an undefined state of the CPU and generate the non maskable interrupt request or the system reset to return the normal state from the undefined state.
Enabling or disabling the WDT operation or the WDT operating clock is selectable by the code option.
See Chapter 26 "Code Option" for details about the code option, See Chapter 5 "Interrupts" for details about the interrupt, and see Chapter 3 "Reset Function" for details about the WDT reset.

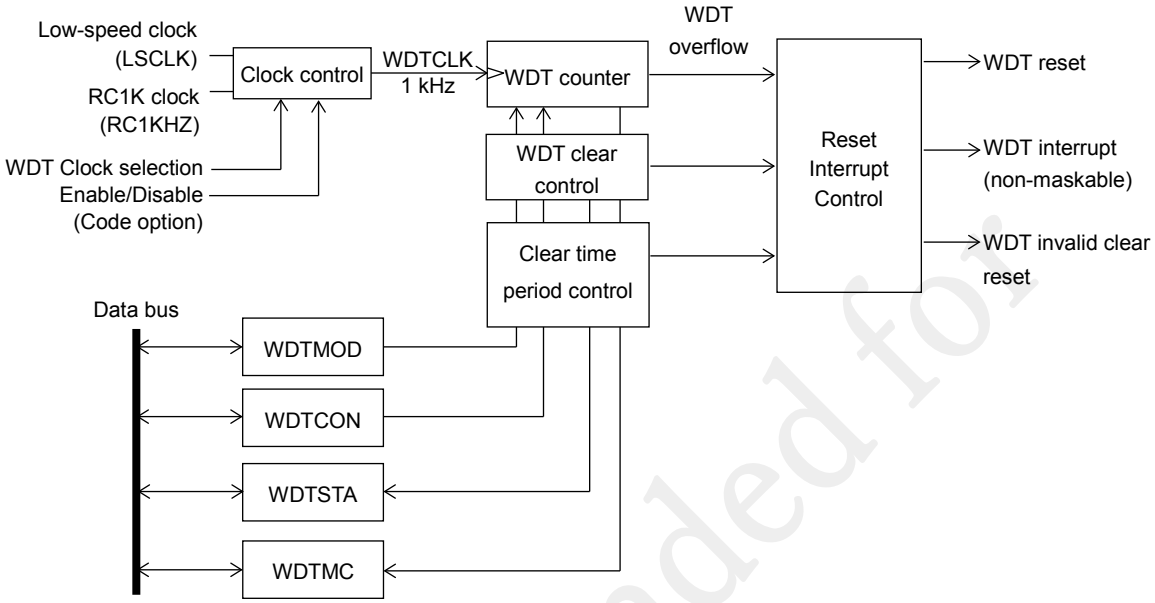
10.1.1 Features

- The operating clock for WDT counter is selectable (LSCLK or RC1KHZ) by the code option
- Eight types of overflow period is selectable (7.8ms, 15.6ms, 31.3ms, 62.5ms, 125ms, 500ms, 2s and 8s) by the software
- Three types of WDT counter clear enable period is selectable by the software
- The first overflow generates the WDT interrupt, and the second overflow generates the WDT reset when the counter clear enable period is 100% of overflow period
- The first overflow generates the WDT reset when the counter clear enable period is 50% or 75% of overflow period
- The WDT invalid clear reset generates when the WDT counter is cleared in the clear disable time period

[Note]
The watchdog timer is for monitoring the CPU's out of control, does not guarantee the function as a general timer.

10.1.2 Configuration

Figure 10-1 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 10-1 Configuration of Watchdog Timer

10.2 Description of Registers

10.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF010	Watchdog timer control register	WDTCON	-	R/W	8	0x00
0xF011	Reserved register	-	-	R	8	0x00
0xF012	Watchdog timer mode register	WDTMOD	-	R/W	8	0x06
0xF013	Reserved register	-	-	R	8	0x00
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 register	-	-	R	8	0x00

[Note]
For the registers with word symbol, word access is possible. For word access, specify an even address.

10.2.2 Watchdog Timer Control Register (WDTCON)

Address: 0xF010
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTCON							
Bit symbol	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	WDP/d0
Access type	W	W	W	W	W	W	W	W	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

WDTCON is a specific function register (SFR) to clear the WDT counter.
When WDTCON is read, the value of the internal pointer (WDP) is read in bit 0.

Description of bits

- **WDP** (Bit 0)
  This is a bit to read the value of the internal pointer (WDP) . The WDP is reset to "0" at the system reset or WDT counter overflow and is reversed every time writing to WDTCON is executed.

- **d7 to d0** (Bit 7 to 0)
  These are bits to write data to clear the WDT counter. The WDT counter can be cleared by writing "0x5A" while the internal pointer (WDP) is "0" and then "0xA5" while the WDP is "1."
  WDT invalid clear reset occurs if clearing the WDT counter during the WDT clear prohibited period.

[Note]
When the interrupt level (ELEVEL) of the U16 core is 2 or higher (during non-maskable interrupt or emulator interrput), the WDT counter cannot be cleared. For more details about ELEVEL, see “nX-U16/100 Core Instruction Manual”.

10.2.3 Watchdog Timer Mode Register (WDTMOD)

Address: 0x0F012
Access: R/W
Access size: 8 bits
Initial value: 0x06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTMOD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	WOVF1	WOVF0	-	WDT2	WDT1	WDT0
Access type	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

WDTMOD is a specific function register (SFR) to set the overflow period of the WDT counter and the WDT clear enable time period.

Description of bits

- **WDT2 to 0** (Bit 2 to 0)
  WDT2, WDT1, and WDT0 are bits to set the overflow period (T_{WOV}) of the WDT counter.

WDT2	WDT1	WDT0	Description
0	0	0	Approx.7.8 ms *1
0	0	1	Approx.15.6 ms *1
0	1	0	Approx. 31.3 ms *1
0	1	1	Approx. 62.5 ms *1
1	0	0	Approx. 125 ms *1
1	0	1	Approx. 500 ms *1
1	1	0	Approx. 2s *1 (initial value)
1	1	1	Approx. 8s *1

*1 : Times when the frequency of WDT counter clock is 1.024kHz sharp.

- **WOVF1 to 0** (Bit 5 to 4)
  These are bits to select the WDT clear enable time period (T_{WCL}).
  When the WDT overflow period 62.5ms or shorter is selected for the overflow period of the WDT counter, the WDT clear enable time period is fixed to 100% regardless the data set in the WOVF1 and WOVF0.

WOVF1	WOVF0	Description
0	0	100 % of the overflow period (initial value)
0	1	75 % of the overflow period
1	0	50% of the overflow period
1	1	

[Note]
When using the counter clear enable period is 50% or 75% of overflow period, the WDT interrupt does not occur. The first overflow generates the WDT reset.

10.2.4 Watchdog Timer Counter Register (WDTMC)

Address: 0x0F014
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	WDTMC															
Byte symbol	WDTMCH								WDTMCL							
Bit symbol	WDTC15	WDTC14	WDTC13	WDTC12	WDTC11	WDTC10	WDTC9	WDTC8	WDTC7	WDTC6	WDTC5	WDTC4	WDTC3	WDTC2	WDTC1	WDTC0
Access type	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

WDTMC is a specific function register (SFR) to read the value of the WDT counter.

Description of bits

- **WDTC15 to WDTC0** (Bit 15 to 0)
  These bits are used to read the counter value of the watch dog timer.
  The operation of WDT counter can be confirmed by periodically reading the bits and checking the data is changed.
  As the count clock of the WDT(WDTCLK) is approx. 1 kHz, read the bits in enouth slower cycle than that by using low speed time base counter interrupt and etc.

[Note]
The count value of the WDT counter is not a consecutive.

10.2.5 Watchdog Status Register (WDTSTA)

Address: 0x0F016
Access: R
Access size: 8 bits
Initial value: 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTSTA							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	WDTCLR2	WDTCLR1	WDTWIN
Access type	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

WDTSTA is a specific function register (SFR) to read the status for clearing the WDT counter.

Description of bits

- **WDTWIN** (Bit 0)
  This bit indicates whether the WDT counter is clearabe or not.

WDTWIN	Description
0	WDT counter clear is disable
1	WDT counter clear is enable (initial value)

- **WDTCLR1** (Bit 1)
  This bit is used to read the reception state of WDT countr clear process.
  It is set to "1" when writing of 0x5A and 0xA5 to the WDTCON register and the WDT clear request is received, and then reset to “0” when the WDT clear process is started.
  Writing to the WDTCON is invalid during the WDTCLR1 bit is “1”. Check if the WDTCLR1 is “0” before writing of 0x5A and 0xA5 to the WDTCON.
  Figure 10-2 shows the timing chart of WDTCLR1.

WDTCLR1	Description
0	WDT counter clear process is not holding (initial value)
1	WDT counter clear process is holding

- **WDTCLR2** (Bit 2)
  This bit is used to read the state of WDT countr clear process.
  It is set to "1" when starting the WDT clear process and reset to “0” when ending the WDT clear process.
  Figure 10-2 shows the timing chart of WDTCLR2.

WDTCLR2	Description
0	WDT counter clear process has not been started (initial value)
1	WDT counter clear process is in progress

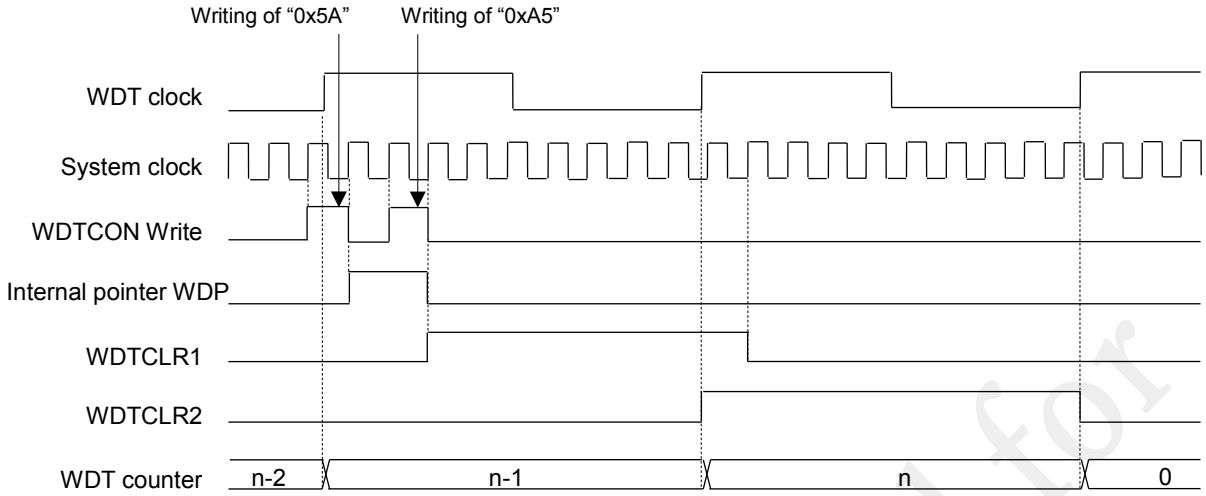


Figure 10-2 Clearing Timing Chart of WDT Counter clear process

[Note]
- Max.two clock of WDTCLK(approx.1KHz) is required between the timing of writing "0x5A" and "0xA5" to the WDTCON and the timing of WDT counter is cleared.
- Confirm if WDTCLR1 is "0" before entering to STOP mode or STOP-D mode.
- Confrim if both WDTCLR1 and WDTCLR2 are "0" just after the WDT clear process, before changin the data of Watch Dog Timer Mode reigster(WDTMOD).

10.3 Description of Operation

Following two operation mode is configurable by the Code Option. See Chapter “26 Code Option” for more details.

- Enable or disable the WDT
- Select the operating clock of WDT counter (WDTCLK) from LSCLK or RC1KHZ.

The WDT counter starts counting up at the rising edge of WDTCLK after the system reset is released.

The WDT counter can be cleared by writing "0x5A" while the internal pointer (WDP) is "0" and then writing "0xA5" while the WDP is "1." The WDP is reset to "0" at the system reset or WDT counter overflow and is reversed every time writing to the WDTCON.

WDT counter overflow period (T_{WOV}) and the WDT counter clear enable time period is selectable by setting watchdog mode register (WDTMOD). Change the WDTMOD just after the WDT clear process, so that unnecessory WDT interrupt or reset do not occur depending on the timing of writing the WDTMOD.
Figure 10-3 shows the program example for changing the mode of Watch Dog Timer.

```
        :
        :
__DI();                 // Disable Interrupts
WDTCON = 0x5a;
WDTCON = 0xa5;          // WDT counter clear process

while ( ( WDTSTA & 0x06 ) != 0x00 ) { }     //   Wait for WDT clear ends

WDTMOD = 0x14;          // Overflow cycle 128ms,   Clear enable time period 75% of overflow cycle
__EI();                 // Enable Interrupts
        :
        :
```

Figure 10-3 Program example for changing the mode of Watch Dog Timer

[Note]

- The watchdog timer cannot detect all the abnormal operations. Even if the CPU loses control, the watchdog timer cannot detect the abnormality in the operation state in which the WDT counter is cleared. It is recommended to clear the WDT counter in a loop of main routine for the fail-safe.
- It’s better to use the RC1KHZ for the WDT conter clock for the fail-safe because the clock independent from the sytem clock, however the RC1KHZ is less accurate than LSCLK. Use the LSCLK if high accurate clock is required.
- The watchdog timer stops in the STOP/STOP-D mode.

10.3.1 Operation when WDT clear enable time period is 100% of overflow cycle

When the WDT counter cannot be cleared within the overflow period (T_{WOV}) , the WDT interrupt occurs. If the WDT counter still remains uncleared and overflows again, the WDT reset occurs.
If the WDT reset occurred, WDTR bit of Reset Status Register (RSTAT) is set to "1" and enters to the system reset mode. See Chapter 3 "Reset Function" for more details about the RSTAT.

Table 10-1 shows the WDT clear enable time period when it is configured as 100% of overflow cycle.

Table 10-1 WDT clear enable time period (when configured as 100% of overflow cycle)

WDT2	WDT1	WDT0	Overflow period (T_{WOV})	WDT reset Period	WDT counter clear enable period (T_{WCL}) *1
0	0	0	Approx.7.8 ms	Approx.15.6 ms	0 to Approx. 13.6 ms
0	0	1	Approx.15.6 ms	Approx.31.3 ms	0 to Approx. 29.3 ms
0	1	0	Approx. 31.3 ms	Approx. 62.5 ms	0 to Approx. 60.5 ms
0	1	1	Approx. 62.5 ms	Approx. 125 ms	0 to Approx. 123 ms
1	0	0	Approx. 125 ms	Approx. 250 ms	0 to Approx. 248 ms
1	0	1	Approx. 500 ms	Approx. 1,000 ms	0 to Approx. 998 ms
1	1	0	Approx. 2,000 ms	Approx. 4,000 ms	0 to Approx. 3998 ms
1	1	1	Approx. 8,000 ms	Approx. 16,000 ms	0 to Approx. 15998 ms

*1 : Times when the frequency of WDT counter clock is 1.024kHz sharp.

Figure 10-4 shows an operation example when the WDT clear enable time period is 100% of overflow cycle.

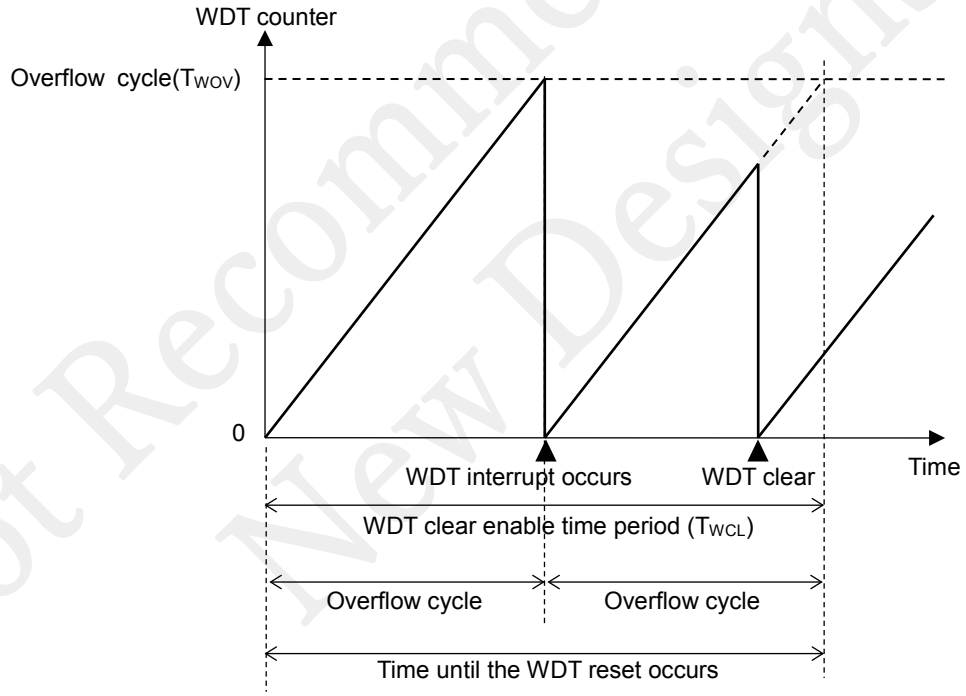


Figure 10-4 WDT operation example (WDT clear enable time is configured as 100% of overflow cycle)

Figure 10-5 shows an operation example of the watchdog timer when the WDT clear enable time period is 100% of overflow cycle.

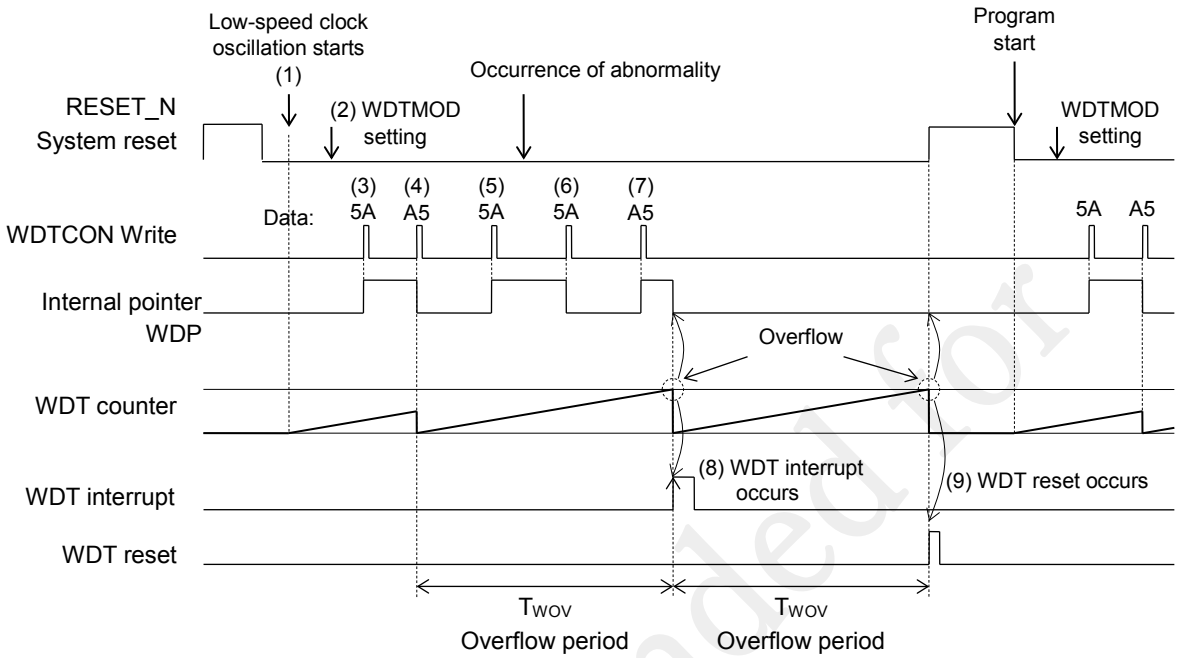


Figure 10-5 Operation Example of Watchdog Timer (when configured as 100% of overflow cycle)

(1) After the system reset is released, the WDT counter starts counting up.
(2) Configure the overflow period (T_{WOV}) of the WDT counter and the WDT clear enable time period to the WDTMOD.
(3) “0x5A” is written to WDTCON. (Internal pointer WDP: 0 ->1)
(4) “0xA5” is written to WDTCON and the WDT counter is cleared. (Internal pointer WDP: 1 ->0)
(5) “0x5A” is written to WDTCON. (Internal pointer WDP: 0 ->1)
(6) When "0x5A" is written in WDTCON after abnormality occurs, it is not accepted because the internal pointer WDP is "1." (Internal pointer WDP: 1 ->0)
(7) "0xA5" is written in WDTCON, however, 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 occurs. (Internal pointer WDP: 1 ->0)
(9) When the WDT counter is not cleared even by software precessing after the WDT interrupt and the WDT counter is overflows again, a WDT reset occurs to enter the system reset mode.

10.3.2 Operation when WDT clear enable time period is 75% or 50% of overflow cycle

When the WDT counter cannot be cleared within the overflow period (T_{WOV}) , the WDT interrupt occurs. If the WDT counter still remains uncleared and overflows again, the WDT reset occurs.
If the WDT reset occurred, WDTR bit of Reset Status Register (RSTAT) is set to "1" and enters to the system reset mode.
If the WDT counter is cleared during the WDT clear prohibited period, the WDT invalid clear reset occurs and WDTWR bit of RSTAT is set to "1" and enters to the system reset mode.
See Chapter 3 "Reset Function" for more details about the RSTAT.

Table 10-2 and 10-3 shows the WDT clear enable time period when it is configured as 75% and 50% of overflow cycle. When the WDT overflow period 62.5ms or shorter is selected for the overflow period of the WDT counter, the WDT clear enable time period is fixed to 100% regardless the data set in the WOVF1 and WOVF0.

Table 10-2 WDT clear enable time period (when configured as 75% of overflow cycle)

WDT2	WDT1	WDT0	Overflow period (T_{WOV})	WDT reset Period	WDT counter clear enable period (window open period: T_{WCL})
0	0	0	Approx.7.8 ms	Approx.7.8 ms	0 to Approx. 5.8 ms (Fixed to 100%)
0	0	1	Approx.15.6 ms	Approx.15.6 ms	0 to Approx. 13.6 ms (Fixed to 100%)
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	0 to Approx. 29.2 ms (Fixed to 100%)
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	0 to Approx. 60.5 ms (Fixed to 100%)
1	0	0	Approx. 125 ms	Approx. 125 ms	Approx. 33 ms to Approx. 123 ms
1	0	1	Approx. 500 ms	Approx. 500 ms	Approx. 126 ms to Approx. 498 ms
1	1	0	Approx. 2,000 ms	Approx. 2,000 ms	Approx. 501 ms to Approx. 3,998 ms
1	1	1	Approx. 8,000 ms	Approx. 8,000 ms	Approx. 2,001 ms to Approx. 7,998 ms

Table 10-3 WDT clear enable time period (when configured as 50% of overflow cycle)

WDT2	WDT1	WDT0	Overflow period (T_{WOV})	WDT reset Period	WDT counter clear enable period (window open period: T_{WCL})
0	0	0	Approx.7.8 ms	Approx.7.8 ms	0 to Approx. 5.8 ms (Fixed to 100%)
0	0	1	Approx.15.6 ms	Approx.15.6 ms	0 to Approx. 13.6 ms (Fixed to 100%)
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	0 to Approx. 29.2 ms (Fixed to 100%)
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	0 to Approx. 60.5 ms (Fixed to 100%)
1	0	0	Approx. 125 ms	Approx. 125 ms	Approx. 64 ms to Approx. 123 ms
1	0	1	Approx. 500 ms	Approx. 500 ms	Approx. 251 ms to Approx. 498 ms
1	1	0	Approx. 2,000 ms	Approx. 2,000 ms	Approx. 1,001 ms to Approx. 3,998 ms
1	1	1	Approx. 8,000 ms	Approx. 8,000 ms	Approx. 4,001 ms to Approx. 7,998 ms

[Note]
- When selecting the 75% or 50% of overflow period as the WDT clear enable time period, the WDT interrupt does not occur, however define the WDT interrupt service function in the program source code. For fale-safe, it is recommended to make program codes that forcely generate the WDT invalid clear reset.
- When selecting the configuration of the overflow cycle is 125ms or longer, use the low-speed oscillation clcok for the WDT count clock by setting the Code Option. The WDT dedicated clock RC1KZ can not be used as the frequency is not accurate.

Figure 10-6 and 10-7 show the timing image when the 75% and 50% of overflow period is configured

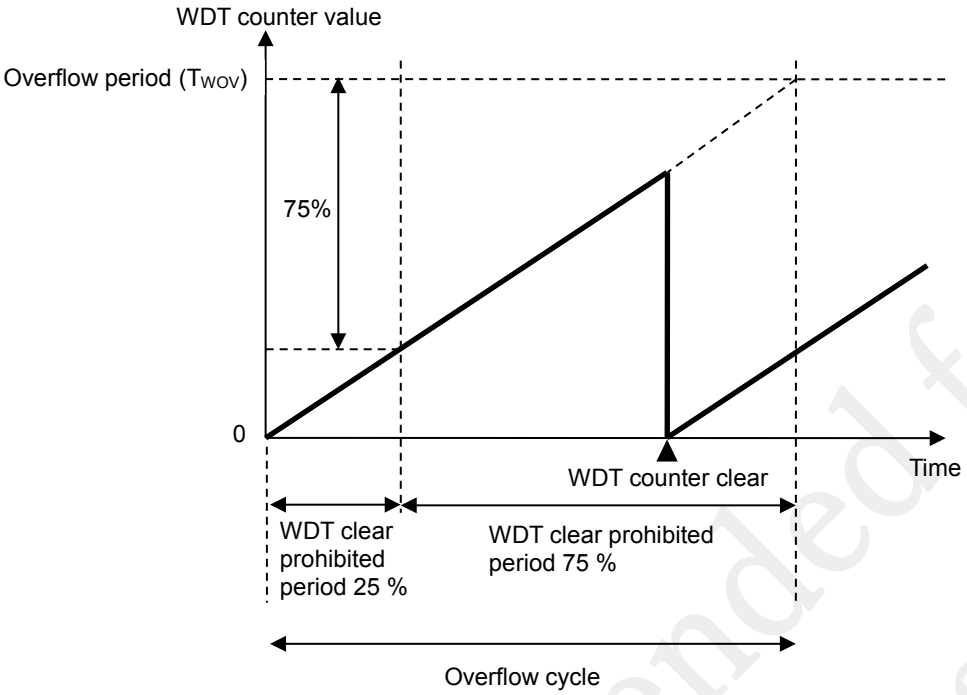


Figure 10-6 WDT operation example (WDT clear enable time is configured as 75% of overflow cycle)

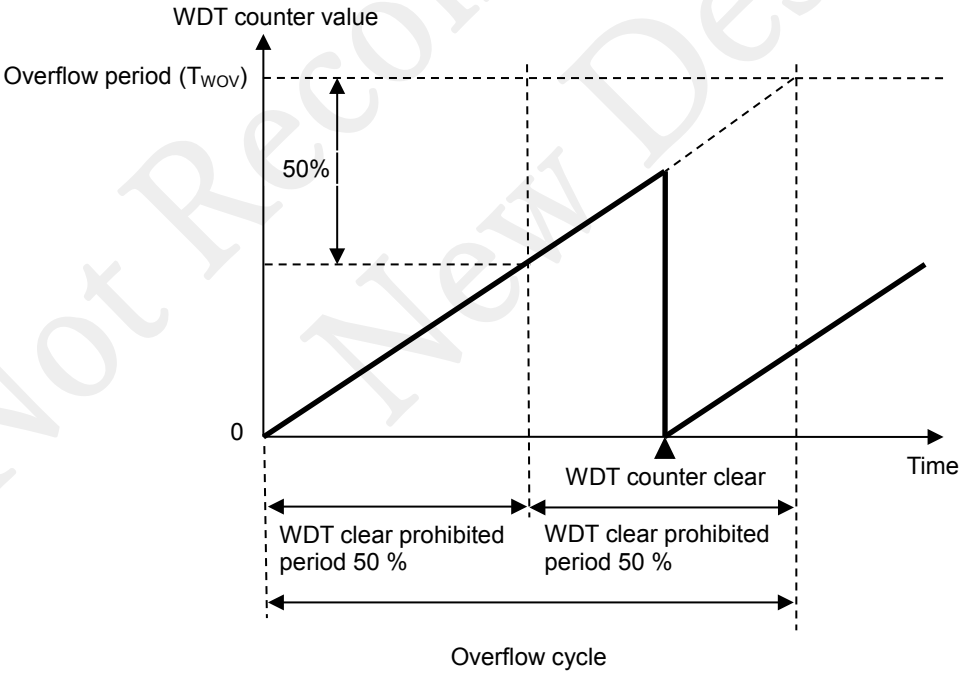


Figure 10-7 WDT operation example (WDT clear enable time is configured as 50% of overflow cycle)

Figure 10-8 shows an operation example of the watchdog timer when the WDT clear enable time period is 75% and 50% of overflow cycle.

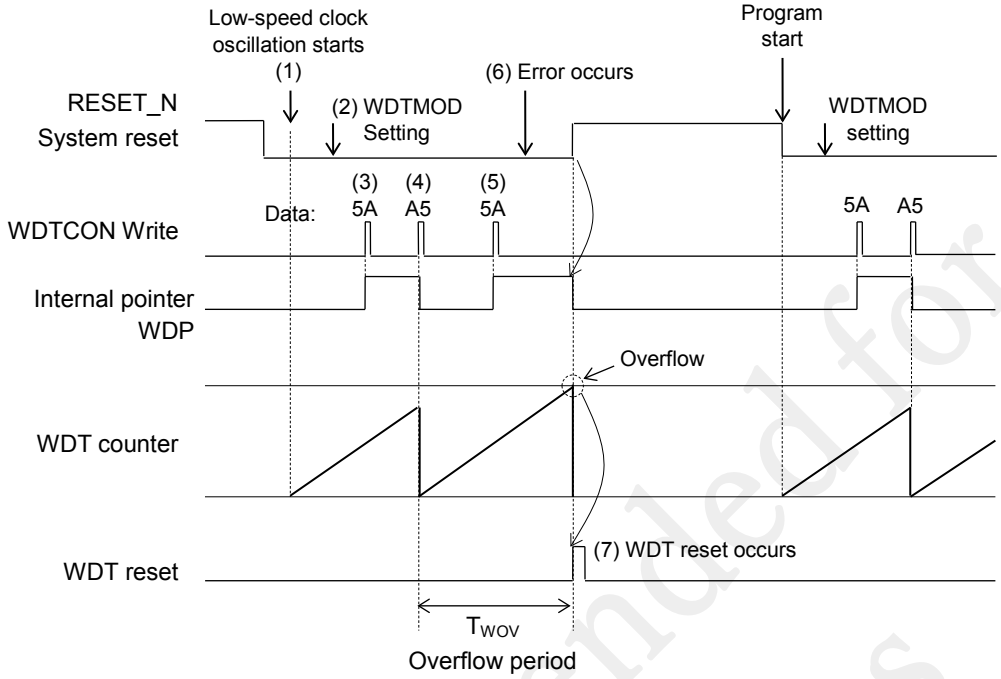


Figure 10-8 Operation Example of Watchdog Timer (Window Open Period 50/75 %)

(1) After the system reset is released, the WDT counter started counting up.
(2) The overflow period (T_{WOV}) of the WDT counter and the window open period of the WDT counter clear window are set in WDTMOD.
(3) "5AH" is written in WDTCON during the open period of the WDT counter clear window. (Internal pointer WDP: 0 ->1)
(4) "0A5H" is written in WDTCON during the open period of the WDT counter clear window and the WDT counter is cleared. (Internal pointer WDP: 1 ->0)
(5) “5AH” is written in WDTCON during the open period of the WDT counter clear window. (Internal pointer WDP: 0 ->1)
(6) Occurrence of abnormality
(7) The WDT counter overflows and a WDT reset occurs. (Internal pointer WDP: 1 ->0)

[Note]
- As the interrupt level (ELEVEL) of the Program Status Word(PSW) in the CPU is set to “2” in the WDT interrupt service routine, the WDT counter cannot be cleared. Clear the WDT counter when the ELEVEL is “0” or “1”.
- It is recommended to clear the WDT counter in a loop of main routine for the fail-safe.

Chapter 11 Serial Communication Unit

11. Serial Communication Unit

11.1 General Description

ML62Q1000 series has two channels of serial communication function : an 8-bit/16-bit synchronous serial port (SSIO) and an asynchronous serial interface UART (Universal Asynchronous Receiver Transmitter).

11.1.1 Features

- Synchronous Serial I/O port (SSIO) mode or Universal Asynchronous Receiver Transmitter(UART) mode is selectable for each channel
- Channel: 2ch
- Synchronous Serial I/O port (SSIO) mode
  - Master/slave modeselectable
  - MSB/LSB first selectable
  - 8-bit/16-bit data length selectable
  - Self-test function using the master and slave modes
    For the self-test functions, see Chapter 29 "Safety Function."
- Asynchronous Receiver Transmitter(UART) mode
  - 5-bit/6-bit/7-bit/8-bit data length selectable
  - Odd parity/even parity/0 parity/1 parity/and no parity selectable
  - 1 stop bit/2 stop bit selectable
  - Positive logic/negative logic selectable for communication logic
  - MSB/LSB first selectable
  - A wide range of communication speed settable
    With clock frequency of 32.768 kHz: 1 to4800 bps
    With clock frequency of 24 MHz: 600 bps to 3 Mbps
    With clock frequency of 16 MHz or 32 MHz: 300 bps to 2 Mbps
  - Built-in baud rate generator for each channel
  - One channel is usable as two channels of half-duplex communication UART
  - Parity error flag, overrun error flag, framing error flag, transmit buffer status flag, receive buffer status flag embedded
  - Self-test function using transmission and reception
    For the self-test function, see Chapter 29 "Safety Function."

11.1.2 Configuration

Figure 11-1 shows the configuration of the serial communication unit.

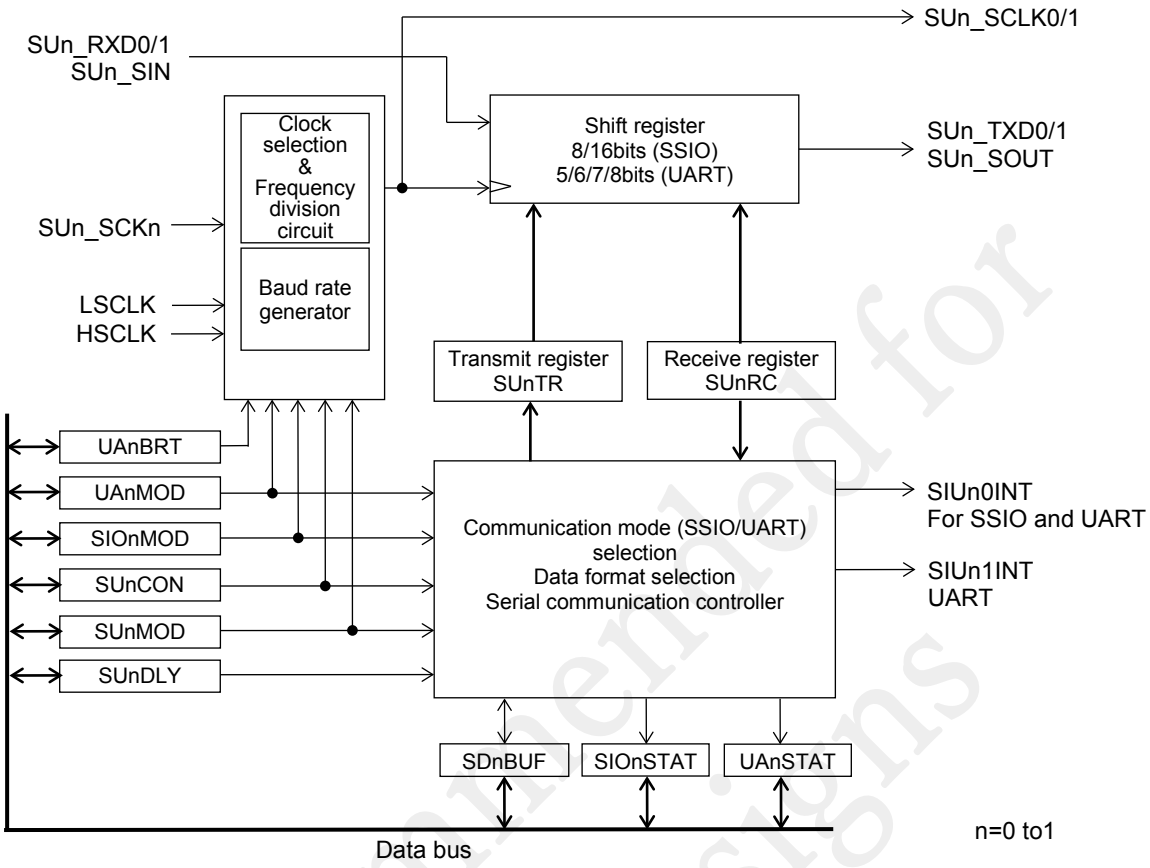


Figure 11-1 Configuration of Serial Communication Unit

SDnBUF : Serial communication unit n transmit/receive buffer
SUnMOD : Serial communication unit n mode register
SUnCON : Serial communication unit n control register
SUnDLY : Serial communication unit n transmission interval setting register
SIOnMOD : SIOn mode register
SIOnSTAT : SIOn status register
UAnMOD : UARTn mode register
UAnBRT : UARTn baud rate register
UAnSTAT : UARTn status register

(n=0 to 1)

11.1.3 List of Pins

The input/output pins of the serial communication unit are assigned to the secondary to octic functions of GPIO. For details of pin assignment and the second to octic settings of GPIO, see Chapter 17 "GPIO."

Pin name	I/O	Function
SU0_RXD0	I	UART0 data input of serial communication unit 0
SU0_RXD1	I	UART1 data input of serial communication unit 0
SU0_TXD0	O	UART0 data output of serial communication unit 0
SU0_TXD1	O	UART1 data output of serial communication unit 0
SU0_SCLK	I/O	SSIO synchronous clock input/output of serial communication unit 0
SU0_SOUT	O	SSIO transmit data output of serial communication unit 0
SU0_SIN	I/	SSIO receive data input of serial communication unit 0
SU1_RXD0	I	UART0 data input of serial communication unit 1
SU1_RXD1	I	UART1 data input of serial communication unit 1
SU1_TXD0	O	UART0 data output of serial communication unit 1
SU1_TXD1	O	UART1 data output of serial communication unit 1
SU1_SCLK	I/O	SSIO synchronous clock input/output of serial communication unit 1
SU1_SOUT	O	SSIO transmit data output of serial communication unit 1
SU1_SIN	I/	SSIO receive data input of serial communication unit 1

11.2 Description of Registers

11.2.1 List of Registers

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF600	Serial communication unit 0 transmit/receive buffer	SD0BUFL	SD0BUF	R/W	8/16	0x00
0xF601		SD0BUFH		R/W	8	0x00
0xF602	Serial communication unit 0 mode register	SU0MOD	-	R/W	8	0x00
0xF603	Reserved register	-	-	R	8	0x00
0xF604	Serial communication unit 0 transmission interval setting register	SU0DLYL	SU0DLY	R/W	8	0x00
0xF605		-		R	8	0x00
0xF606	Serial communication unit 0 control register	SU0CONL	SU0CON	R/W	8/16	0x00
0xF607		SU0CONH		R/W	8	0x00
0xF608	Synchronous serial port 0 mode register	SIO0MODL	SIO0MOD	R/W	8/16	0x00
0xF609		SIO0MODH		R/W	8	0x00
0xF60A	Synchronous serial port 0 status register	SIO0STAT	-	R/W	8	0x00
0xF60B	Reserved register	-	-	R	8	0x00
0xF60C	UART00 mode register	UA00MODL	UA00MOD	R/W	8/16	0x00
0xF60D		UA00MODH		R/W	8	0x00
0xF60E	UART00 baud rate register	UA00BRTL	UA00BRT	R/W	8/16	0xFF
0xF60F		UA00BRTH		R/W	8	0xFF
0xF610	UART00 baud rate adjustment register	UA00BRC	-	R/W	8	0x00
0xF611	Reserved register	-	-	R	8	0x00
0xF612	UART00 status register	UA00STAT	-	R/W	8	0x00
0xF613	Reserved register	-	-	R	8	0x00
0xF614	UART01 mode register	UA01MODL	UA01MOD	R/W	8/16	0x00
0xF615		UA01MODH		R/W	8	0x00
0xF616	UART01 baud rate register	UA01BRTL	UA01BRT	R/W	8/16	0xFF
0xF617		UA01BRTH		R/W	8	0xFF
0xF618	UART01 baud rate adjustment register	UA01BRC	-	R/W	8	0x00
0xF619	Reserved register	-	-	R	8	0x00
0xF61A	UART01 status register	UA01STAT	-	R/W	8	0x00
0xF61B	Reserved register	-	-	R	8	0x00

Address	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value
0xF620	Serial communication unit 1 transmit/receive buffer	SD1BUFL	SD1BUF	R/W	8/16	0x00
0xF621		SD1BUFH		R/W	8	0x00
0xF622	Serial communication unit 1 mode register	SU1MOD	-	R/W	8	0x00
0xF623	Reserved register	-	-	R	8	0x00
0xF624	Serial communication unit 1 transmission interval setting register	SU1DLYL	SU1DLY	R/W	8	0x00
0xF625		-		R	8	0x00
0xF626	Serial communication unit 1 control register	SU1CONL	SU1CON	R/W	8	0x00
0xF627		SU1CONH		R	8	0x00
0xF628	Synchronous serial port 1 mode register	SIO1MODL	SIO1MOD	R/W	8/16	0x00
0xF629		SIO1MODH		R/W	8	0x00
0xF62A	Synchronous serial port 1 status register	SIO1STAT	-	R/W	8	0x00
0xF62B	Reserved register	-	-	R	8	0x00
0xF62C	UART10 mode register	UA10MODL	UA10MOD	R/W	8/16	0x00
0xF62D		UA10MODH		R/W	8	0x00
0xF62E	UART10 baud rate register	UA10BRTL	UA10BRT	R/W	8/16	0xFF
0xF62F		UA10BRTL	-	R/W	8	0xFF
0xF630	UART10 baud rate adjustment register	UA10BRC	-	R/W	8	0x00
0xF631	Reserved register	-	-	R	8	0x00
0xF632	UART10 status register	UA10STAT	-	R/W	8	0x00
0xF633	Reserved register	-	-	R	8	0x00
0xF634	UART11 mode register	UA11MODL	UA11MOD	R/W	8/16	0x00
0xF635		UA11MODH		R/W	8	0x00
0xF636	UART11 baud rate register	UA11BRTL	UA11BRT	R/W	8/16	0xFF
0xF637		UA11BRTH		R/W	8	0xFF
0xF638	UART11 baud rate adjustment register	UA11BRC	-	R/W	8	0x00
0xF639	Reserved register	-	-	R	8	0x00
0xF63A	UART11 status register	UA11STAT	-	R/W	8	0x00
0xF63B	Reserved register	-	-	R	8	0x00

[Note]
For the registers with word symbol, word access is possible. For word address, specify an even address.

Table 11-1 shows the SFRs, Communication pins and Interrupts used in each communication mode.
The communication mode is selectable by setting SUnMD1 bit and SUnMD0 bit of the serial communication unit n mode register (SUnMOD).

Table 11-1 List of SFRs, Communication pins and Interrupts

Item	SFR Name	SFR symbol name (Byte)	SSIO	UART Full duplex mode	UART Half duplex mode	
					UARTn1	UARTn0
SFR	Serial Communication Unit n Transmit/Receive buffer	SDnBUFL	○	○ Use as a receive buffer	—	○
		SDnBUFH	○ Use in 16bit mode	○ Use as a transmit buffer	○	—
	Serial Communication Unit n Mode Register	SUnMOD	○	○	○	○
	Serial Communication Unit n Transmission Interval Setting Register	SUnDLYL	○	○ Only transmit	○	○
	Serial Communication Unit n Control Register	SUnCONL	○ Only SnEN	○ Only Un0EN	—	○ Only Un0EN
		SUnCONH	—	—	○ Only Un1EN	—
	Synchronous Serial Port n Mode Register	SIOnMODL SIOnMODH	○	—	—	—
	Synchronous Serial Port n Status Register	SIOnSTAT	○	—	—	—
	UARTn0 Mode Register	UAn0MODL UAn0MODH	—	○	—	○
	UARTn0 Baud Rate Register	UAn0BRTL UAn0BRTH	—	○	—	○
	UARTn0 Baud Rate Adjustment Register	UAn0BRC	—	○	—	○
	UARTn0 Status Register	UAn0STAT	—	○	—	○
	UARTn1 Mode Register	UAn1MODL UAn1MODH	—	—	○	—
	UARTn1 Baud Rate Register	UAn1BRTL UAn1BRTH	—	—	○	—
	UARTn1 Baud Rate Adjustment Register	UAn1BRC	—	—	○	—
	UARTn1 Status Register	UAn1STAT	—	—	○	—
SSIO pin	Data input pin	—	SUn_SIN	—	—	—
	Data output pin	—	SUn_SOUT	—	—	—
	Clock in/out pin	—	SUn_SCLK	—	—	—
UART pin	RXD pin	—	—	SUn_RXD0	SUn_RXD1	SUn_RXD0
	TXD pin	—	—	SUn_TXD1	SUn_TXD1	SUn_TXD0
Interrupt	Receive interrupt	—	SIUn0INT	SIUn0INT	SIUn1INT	SIUn0INT
	Transmit interrupt	—		SIUn1INT		

○: Used -: Unused

11.2.2 Serial Communication Unit n Transmit/Receive Buffer (SDnBUF)

Address: 0xF600, 0xF620
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SDnBUF															
Byte symbol	SDnBUFH								SDnBUFL							
Bit symbol	SnB15	SnB14	SnB13	SnB12	SnB11	SnB10	SnB9	SnB8	SnB7	SnB6	SnB5	SnB4	SnB3	SnB2	SnB1	SnB0
Access type	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 serial communication unit n transmit/receive buffer (SDnBUF) is a specific function register (SFR) to write transmit data and read receive data of the serial communication unit.

The buffers are used in different way in each communication mode.

Communication mode		SDnBUF	
		SDnBUFH	SDnBUFL
SSIO mode	8-bit mode	Unused	Transmit/receive buffer
	16-bit mode	Transmit/receive buffer (higher 8 bits)	Transmit/receive buffer (lower 8 bits)
UART mode	Full-duplex communication mode	UARTn transmit buffer	UARTn receive buffer
	Half-duplex communication mode	UARTn1 transmit/receive buffer	UARTn0 transmit/receive buffer

- Synchronous Serial Port (SSIO) mode

By writing a transmit data to the SDnBUF, the data is stored into the transmit register (SUnTR) too. A data in the receive register (SUnRC) is read by reading the SDnBUF. The SDnBUFH is not used in the 8-bit mode.

- Full-duplex UART communication mode

SDnBUFL is used as the receive buffer and SDnBUFH is used as the transmit buffer.

In the receive operation, a received data is overwritten to SDnBUFL when the reception is completed. Read the SDnBUFL after the UARTn interrupt occurred at the end of receive. Writing to SDnBUFL is invalid in the Full-duplex UART communication mode. When a data length of 5 to 7-bit length is selected, "0" is read from an unused bit.

In the transmit operation, write a transmit data into the SDnBUFH. To transmit data successively, check that the Un1FUL bit of the UARTn1 status register (UAn1STAT) is "0", then write the next transmit data into the SDnBUFH. The SDnBUFH is readable. When a data length of 5 to 7-bit length is selected, the data written in an unused bit is invalid.

- Half-duplex UART communication mode

SDnBUFL and SDnBUFH is used as receive/transmit buffer in each channel.

In the receive operation, a received data is overwritten to SDnBUFL and SDnBUFH when the reception is completed. Read the SDnBUFL after the UARTn interrupt occurred at the end of receive. Writing to SDnBUFL and SDnBUFH is invalid in the Half-duplex UART communication mode. When a data length of 5 to 7-bit length is selected, "0" is read from an unused bit.

In the transmit operation, write a transmit data into the SDnBUFH and SDnBUFH. To transmit data successively, check that the Un0FUL/Un1FUL bit of the UARTn0/UARTn1 status register (UAn0STAT/UAn1STAT) is "0", then write the next transmit data into the SDnBUFL and SDnBUFH. The SDnBUFL and SDnBUFH is readable. When a data length of 5 to 7-bit length is selected, the data written in an unused bit is invalid.

[Note]
- In the half-duplex communication mode of UART, be sure to select the transmit mode by setting Un0IO and Un1IO bit of the UARTn mode register (UAn0MOD, UAn1MOD) before writing the transmit data in SDnBUFL and SDnBUFH.
- Do not perform write-operation to the SDnBUF in the SSIO slave receive mode.

11.2.3 Serial Communication Unit n Mode Register (SUnMOD)

Address: 0xF602, 0xF622
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								SUnMOD							
Bit symbol	-	-	-	-	-	-	-	-	-	SUnTIMD	SUnRIMD	-	-	-	SUnMD1	SUnMD0
Access type	R	R	R	R	R	R	R	R	R	R/W	R/W	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

The serial communication unit n mode register (SUnMOD) is a specific function register (SFR) to select the communication mode of the serial communication unit.

Description of bits

- **SUnMD1, SUnMD0** (Bit 1 to 0)
  SUnMD1 and SUnMD0 are bits to select the communication mode of the serial communication unit.

SUnMD1	SUnMD0	Description
0	0	SSIO mode (initial value)
0	1	SSIO mode
1	0	UART full-duplex communication mode
1	1	UART half-duplex communication mode × 2 channels

- **SUnRIMD** (Bit 5)
  SUnRIMD is a bit to select the timing of receive interrupt occurrence.

SUnRIMD	Description
0	The Interrupt occurs at the end of data reception (initial value)
1	The Interrupt occurs at the start and end of data reception

- **SUnTIMD** (Bit 6)
  SUnTIMD is a bit to select the timing of transmit interrupt occurrence.

SUnTIMD	Description
0	The Interrupt occurs at the end of data transmission (initial value)
1	The Interrupt occurs at the start and end of data transmission

[Note]
Be sure to set the SUnMOD register while communication is stopped (SUnCON=0x00). Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

11.2.4 Serial Communication Unit n Transmission Interval Setting Register (SUnDLY)

Address: 0xF604, 0xF624
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SUnDLY															
Byte symbol	-								SUnDLYL							
Bit symbol	-	-	-	-	-	-	-	-	SUnDLY7	SUnDLY6	SUnDLY5	SUnDLY4	SUnDLY3	SUnDLY2	SUnDLY1	SUnDLY0
Access type	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

SUnDLY is a specific function register (SFR) to set the transmit frame interval of serial communication. It is used to wait until the slave device receives data and processes it when serial data is transmitted in continuous communication.

Description of bits

- **SUnDLY7 to 0** (Bit 7 to 0)

  SUnDLY7 to 0 is for setting the transmit frame interval in the SSIO mode and UART mode.
  When SUnDLY is "0x00", the transmit frame interval of serial communication is "0".
  When SUnDLY is not "0x00", the transmit frame interval of serial communication is set in the following formula.

  SSIO mode:
  [Transmit frame interval of serial communication] =
  Transfer clock cycle[*1] x ROUNDUP[*2]((SUnDLYL+2) / (frequency division value of HSCLK))

  UART mode:
  [Transmit frame interval of serial communication] =
  Base clock cycle[*3] x (UAn0BRT+1) x ROUNDUP[*2]((SUnDLYL+2) / (UAn0BRT+1))

  [*1] Transfer clock cycle = (frequency division value of HSCLK selected by SnCK4-0 bits) x (cycle of HSCLK)
  [*2] ROUNDUP means the mathmatical operation of rounding a number up to the next higher integer.
  [*3] Base clock cycle = The cycle of base clock selected by UAn0MOD and UAn1MOD.

[Note]
- Set "0x00" to the SUnDLYL register in the SSIO slave mode.
- The SUnDLYL register is disabled in the SSIO master receive mode.

11.2.5 Serial Communication Unit n Control Register (SUnCON)

Address: 0xF606, 0xF626
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 symbol	SUnCON															
Byte symbol	SUnCONH								SUnCONL							
Bit symbol	-	-	-	-	-	-	Un1EN	-	-	-	-	-	-	-	Un0EN	SnEN
Access type	R	R	R	R	R	R	R/W	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

SUnCON is a specific function register (SFR) to control the serial communication unit.

Description of bits

- **SnEN** (Bit 0)
  SnEN is a bit to enable the synchronous serial communication in the SSIO mode.
  Reset the bit to "0" when terminating the communication.
  SnEN bit is not writable in the UART mode.

SnEN	Description
0	Stop communication (initial value)
1	Start communication

- **Un0EN** (Bit 1)
  Un0EN is a bit to enable the UARTn communication in the UART mode. Un0EN bit is not writable in the UART mode.

Un0EN	Description	
	Full-duplex mode	Half-duplex mode
0	Stop UARTn communication (initial value)	Stop UARTn0 communication
1	Start UARTn communication	Start UARTn0 communication

- **Un1EN** (Bit 9)
  Un1EN is a bit to enable the UARTn1 communication in the half-duplex mode.
  Un1EN is not writable in the full-duplex UART mode or SSIO mode.
  .

Un1EN	Description	
	Full-duplex mode	Half-duplex mode
0	Unused	Stop UARTn1 communication (initial value)
1	Unused	Start UARTn1 communication

11.2.6 Synchronous Serial Port n Mode Register (SIOnMOD)

Address: 0xF608, 0xF628
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SIOnMOD															
Byte symbol	SIOnMODH								SIOnMODL							
Bit symbol	-	SnNEG	SnCKT	SnCK4	SnCK3	SnCK2	SnCK1	SnCK0	-	-	-	-	SnLG	SnMD1	SnMD0	SnDIR
Access type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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

SIOnMOD is a specific function register (SFR) to set the communication mode when SSIO is selected.

Description of bits

- **SnDIR** (Bit 0)
  SnDIR is a bit to select the communication direction in the SSIO mode.

S0DIR	Description
0	LSB first (initial value)
1	MSB first

- **SnMD1, SnMD0** (Bit 2 to 1)
  SnMD1 and SnMD0 are bits to select the transmit/receive mode in the SSIO mode.

SnMD1	SnMD0	Description
0	0	Stop transmission/reception (initial value)
0	1	Receive mode
1	0	Transmit mode
1	1	Transmit/receive mode

- **SnLG** (Bit 3)
  SnLG is a bit to select the bit length of the transmit/receive buffer when the SSIO mode is selected.

SnLG	Description
0	8-bit length (initial value)
1	16-bit length

- **SnCK4 to SnCK0** (Bit 12 to 8)
  SnCK4 to SnCK0 are bits to select the transfer clock of SSIO.
  When the internal clock is selected for the transfer clock, SSIO enters the master mode. When the external clock is selected, it enters the slave mode. For the frequency of external clock in the slave mode, see the data sheet of each product.

SnCK4	SnCK3	SnCK2	SnCK1	SnCK0	Description
0	0	0	0	0	1/1 LSCLK (initial value)
0	0	0	0	1	1/2 LSCLK
1	0	0	0	0	1/1 HSCLK
1	0	0	0	1	1/2 HSCLK
1	0	0	1	0	1/4 HSCLK
1	0	0	1	1	1/8 HSCLK
1	0	1	0	0	1/16 HSCLK K
1	0	1	0	1	1/32 HSCLK
1	0	1	1	0	1/64 HSCLK
1	0	1	1	1	1/128 HSCLK
1	1	0	0	0	External clock (Slave mode)
Others					Reserved register (1/1 LSCLK)

- **SnCKT** (Bit 13)
  SnCKT is a bit to select the phase of transfer clock output in the SSIO mode.
  When combined with SnNEG, it allows to select four types of communication.

SnCKT	Description
0	Clock type 0: Output with Initial value = "H" level (initial value).
1	Clock type 1: Output with Initial value = "L" level.

- **SnNEG** (Bit 14)
  SnNEG is a bit to select the logic of transfer clock in the SSIO mode.

SnNEG	Description
0	Positive logic (initial value)
1	Negative logic

[Note]
- Be sure to set the SIO0MOD register while communication is stopped. Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.
- Set the S0CK4 to S0CK0 bits to 4MHz or below.
- Enable the high-speed oscillation when selecting the slave mode. See Chapter 6 “Clock Generation Circuit” for details on how to enable the high-speed oscillation.
- The maximum frequency of communication clock is 1MHz in the slave mode.
- Specify the frequency of transfer clock as 24MHz or lower.

11.2.7 Synchronous Serial Port n Status Register (SIOnSTAT)

Address: 0xF60A, 0xF62A
Access: R
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								SnSTAT							
Bit symbol	-	-	-	-	-	-	-	-	-	-	SnRXF	SnTXF	SnFUL	-	-	-
Access type	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

SnSTAT is s specific function register (SFR) to indicate the state in the transmit/receive operation of SSIO when the SSIO mode is selected.

Description of bits

- **SnFUL** (Bit 3)
  SnFUL is a bit to indicate the transfer state of the transmit buffer (SDnBUF) when the transmit mode of the SSIO mode is selected.
  SnFUL bit is set to “1” by writing a data to SDnBUF and reset to “0” when starting to transfer the data.
  When the SnEN bit of the SUnCON register is set to "1" on the condition of SnFUL is “1”, the transmission starts. If the SnEN bit of the SUnCON register is set to "1" on the condition of SnFUL is “0”, the transmission does not starts until a data is written to the SDnBUF.
  If writing data to SDnBUF on the condtion of SnFUL is “1”, the data of SDnBUF is overwritten.

SnFUL	Description
0	Transmit buffer has no data (initial value)
1	Transmit buffer has data

- **SnTXF** (Bit 4)
  SnTXF is a bit to indicate that SSIO data is in transmitting in the SSIO mode.

SnTXF	Description
0	Data transmission stopped (initial value)
1	Data transmission in progress

- **SnRXF** (Bit 5)
  SnRXF is a bit to indicate that SSIO data is in receiving in the SSIO.

SnRXF	Description
0	Data reception stopped (initial value)
1	Data reception in progress

11.2.8 UARTn0 Mode Register (UAn0MOD)

Address: 0xF60C, 0xF62C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	UAn0MOD															
Byte symbol	UAn0MODH								UAn0MODL							
Bit symbol	Un0DIR	Un0NEG	Un0STP	Un0PT2	Un0PT1	Un0PT0	Un0LG1	Un0LG0	Un0RSS	-	-	-	-	Un0CK1	Un0CK0	Un0IO
Access type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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

UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communiction mode and half-duplex communication mode.

Description of bits

- **Un0IO** (Bit 0)
  Un0IO is a bit to select the transmit mode or receive mode in UARTn0 full-duplex and half-duplex mode..
  When the full-duplex communication mode is selected, the Un0IO bit is fixed to "1" to set to the receive mode.

Un0IO	Description
0	Transmit mode (initial value)
1	Receive mode

- **Un0CK1, Un0CK0** (Bit 2 to 1)
  Un0CK1 and Un0CK0 are bits to select the base clock of baud rate generator in UARTn0 full-duplex and half-duplex mode.

Un0CK1	Un0CK0	Description
0	0	LSCLK (initial value)
0	1	Do not use (LSCLK)
1	0	HSCLK
1	1	Do not use (HSCLK)

- **Un0RSS** (Bit 7)
  Un0RSS is a bit to select the sampling timing of the receive data in UARTn0 full-duplex and half-duplex mode.

Un0RSS	Description
0	Values set to UAn0BRTH and UAn0BRTL registers/2 (initial value)
1	Values set to UAn0BRTH and UAn0BRTL registers/2-1

- **Un0LG1, Un0LG0** (Bit 9 to 8)
  Un0LG1 and Un0LG0 are bits to select the communication data length in UARTn0 full-duplex and half-duplex mode.

Un0LG1	Un0LG0	Description
0	0	8-bit length (initial value)
0	1	7-bit length
1	0	6-bit length
1	1	5-bit length

- **Un0PT2, Un0PT1, Un0PT0** (Bit 12 to 10)
  Un0PT2, Un0PT1, and Un0PT0 are bits to select the parity bit in UARTn0 full-duplex and half-duplex mode.

Un0PT2	Un0PT1	Un0PT0	Description
*	*	0	No parity bit (initial value)
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Parity bit fixed to "1"
1	1	1	Parity bit fixed to "0"

- **Un0STP** (Bit 13)
  Un0STP is a bit to select the stop bit length in UARTn0 full-duplex and half-duplex mode.

Un0STP	Description
0	1 stop bit (initial value)
1	2 stop bit

- **Un0NEG** (Bit 14)
  Un0NEG is a bit to select the logic of data output in UARTn0 full-duplex and half-duplex mode.

Un0NEG	Description
0	Positive logic (initial value)
1	Negative logic

- **Un0DIR** (Bit 15)
  Un0DIR is a bit to select the communication direction in UARTn0 full-duplex and half-duplex mode.

Un0DIR	Description
0	LSB first (initial value)
1	MSB first

[Note]
Be sure to set the UAn0MOD register while communication is stopped. Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

11.2.9 UARTn1 Mode Register (UAn1MOD)

Address: 0xF614, 0xF634
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	UAn1MOD															
Byte symbol	UAn1MOD1								UAn1MOD0							
Bit symbol	Un1DIR	Un1NEG	Un1STP	Un1PT2	Un1PT1	Un1PT0	Un1LG1	Un1LG0	Un1RSS	-	-	-	-	Un1CK1	Un1CK0	Un1IO
Access type	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	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UAn1MOD is a specific function register (SFR) to set the transfer mode in uARTn1 half-duplex communication mode. When full-duplex communication is selected, data in UAn0MOD is undefined.

Description of bits

- **Un1IO** (Bit 0)
  Un1IO is a bit to select the transmit mode or receive mode in UARTn1 half-duplex mode..
  When the full-duplex communication mode is selected, the Un1IO bit is fixed to "1" to set to the receive mode.

Un1IO	Description
0	Transmit mode (initial value)
1	Receive mode

- **Un1CK1, Un1CK0** (Bit 2 to 1)
  Un1CK1 and Un1CK0 are bits to select the base clock of baud rate generator in UARTn1 half-duplex mode.

Un1CK1	Un1CK0	Description
0	0	LSCLK (initial value)
0	1	Reserved (do not use)
1	0	HSCLK
1	1	HSCLK

- **Un1RSS** (Bit 7)
  Un1RSS is a bit to select the sampling timing of the receive data in UARTn1 half-duplex mode.

Un1RSS	Description
0	Values set to UAn1BRTH and UAn1BRTL registers/2 (initial value)
1	Values set to UAn1BRTH and UAn1BRTL registers/2-1

- **Un1LG1, Un1LG0** (Bit 9 to 8)
  Un1LG1 and Un1LG0 are bits to select the communication data length in UARTn1 half-duplex mode.

Un1LG1	Un1LG0	Description
0	0	8-bit length (initial value)
0	1	7-bit length
1	0	6-bit length
1	1	5-bit length

- **Un1PT2, Un1PT1, Un1PT0** (Bit 12 to 10)
  Un1PT2, Un1PT1, and Un1PT0 are bits to select the parity bit in UARTn1 half-duplex mode.

Un1PT2	Un1PT1	Un1PT0	Description
*	*	0	No parity bit (initial value)
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Parity bit fixed to "1"
1	1	1	Parity bit fixed to "0"

- **Un1STP** (Bit 13)
  Un1STP is a bit to select the stop bit length in UARTn1 half-duplex mode.

Un1STP	Description
0	1 stop bit (initial value)
1	2 stop bit

- **Un1NEG** (Bit 14)
  Un1NEG is a bit to select the logic of data in UARTn1 half-duplex mode.

Un1NEG	Description
0	Positive logic (initial value)
1	Negative logic

- **Un1DIR** (Bit 15)
  Un1DIR is a bit to select the communication direction in UARTn1 half-duplex mode.

Un1DIR	Description
0	LSB first (initial value)
1	MSB first

[Note]
Be sure to set the UAn1MOD register while communication is stopped. Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

11.2.10 UART0 Baud Rate Register (UAn0BRT)

Address: 0xF60E, 0xF62E
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	UAn0BRT															
Byte symbol	UAn0BRTH								UAn0BRTL							
Bit symbol	Un0BR15	Un0BR14	Un0BR13	Un0BR12	Un0BR11	Un0BR10	Un0BR9	Un0BR8	Un0BR7	Un0BR6	Un0BR5	Un0BR4	Un0BR3	Un0BR2	Un0BR1	Un0BR0
Access type	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

UAn0BRT is a specific function register (SFR) to set the count value of the baud rate generator in UARTn0 full-duplex communiction mode and half-duplex communication mode. For the relationship between the count value of the baud rate generator and the baud rate, see Chpater "11.3.2.2 Baud Rate."

11.2.11 UARTn1 Baud Rate Register (UAn1BRT)

Address: 0F616H, 0F636H
Access: R/W
Access size: 8/16 bits
Initial value: 0FFFFH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	UAn1BRT															
Byte symbol	UAn1BRTH								UAn1BRTL							
Bit symbol	Un1BR15	Un1BR14	Un1BR13	Un1BR12	Un1BR11	Un1BR10	Un1BR9	Un1BR8	Un1BR7	Un1BR6	Un1BR5	Un1BR4	Un1BR3	Un1BR2	Un1BR1	Un1BR0
Access type	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

UAn1BRT is a specific function register (SFR) to set the count value of the baud rate generator in UARTn0 half-duplex communication mode. No need to set data in UAn1BRT when the full-duplex communication is selected.
For the relationship between the count value of the baud rate generator and the baud rate, see Chpater "11.3.2.2 Baud Rate."

> [Note]
> Be sure to set the UAn0BRT and UAn1BRT register while communication is stopped. Do not rewrite it during communication.

11.2.12 UARTn0 Baud Rate Adjustment Register (UAn0BRC)

Address: 0xF610, 0xF630
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								UAn0BRC							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	Un0BRC2	Un0BRC1	Un0BRC0
Access type	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

UAn0BRC is a specific function register (SFR) to adjust the count value of the baud rate generator in UARTn0 full-duplex communiction mode and half-duplex communication mode.
For the relationship between the value of UAn0BRC and the correction value, see Chpater "11.3.2.2 Baud Rate."

11.2.13 UARTn1 Baud Rate Adjustment Register (UAn1BRC)

Address: 0xF618, 0xF638
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								UAn1BRC							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	Un1BRC2	Un1BRC1	Un1BRC0
Access type	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

UAn1BRC is a specific function register (SFR) to adjust the count value of the baud rate generator in UARTn1 half-duplex communication mode.
No need to set data in UAn1MOD when the full-duplex communication is selected.
For the relationship between the value of UAn1BRC and the correction value, see "11.3.2.2 Baud Rate."

> [Note]
> communication. Be sure to set the UAn0BRC and UAn1BRC register while communication is stopped. Do not rewrite it during communication.

11.2.14 UARTn0 Status Register (UAn0STAT)

Address: 0xF612, 0xF632
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								UAn0STAT							
Bit symbol	-	-	-	-	-	-	-	-	-	-	Un0RXF	Un0TXF	Un0FUL	Un0PER	Un0OER	Un0FER
Access type	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

UAn0STAT is a specific function register (SFR) to indicate the state in the transmit/receive operation in UARTn0 full-duplex communiction mode and half-duplex communication mode.

Description of bits

- **Un0FER** (Bit 0)
  Un0FER is a bit to indicate a framing error in UARTn0 full-duplex communiction mode and half-duplex communication mode.
  The Un0FER becomes "1" when an error occurs in the stop bit and holds “1” until it is cleared by the software.
  When "1" is written in Un0FER, it is rest to "0."
  Un0FER is fixed to "0" in the transmit mode.

Un0FER	Description
0	Framing error does not occur (initial value)
1	Framing error occurs

- **Un0OER** (Bit 1)
  Un0OER is a bit to indicate an overrun error in UARTn0 full-duplex communiction mode and half-duplex communication mode.
  It becomes "1" when data is received again before the receive data of the serial communication unit n transmit/receive buffer (SDnBUFL) is read.
  When "1" is written in Un0OER, it is forcibly rest to "0."
  Un0OER is fixed to "0" in the transmit mode.

Un0OER	Description
0	Overrun error does not occur (initial value)
1	Overrun error occur

- **Un0PER** (Bit 2)
  Un0PER is a bit to indicate a parity error in UARTn0 full-duplex communiction mode and half-duplex communication mode.
  The parity of the received data and the parity bit added to the data are compared. If they do not match, the Un0PER becomes "1" and holds “1” until it is cleared by the software.
  When "1" is written in Un0PER, it is reset to "0."
  Un0PER is fixed to "0" in the transmit mode.

Un0PER	Description
0	Parity error does not occur (initial value)
1	Parity error occurs

- **Un0FUL** (Bit 3)
  Un0FUL is a bit to indicate the state of the transmit/receive buffer in UARTn0 full-duplex communiction mode and half-duplex communication mode.
  When the full-duplex communication mode is selected, it becomes "1" when transmit data is written in SD0BUFH and becomes "0" when the transmit data is transferred to the shift register. To transmit data successively, check that the Un0FUL flag becomes "0", then write the next transmit data in SDn0BUFH.
  When the half-duplex mode is selected, it becomes "1" when transmit data is written in SDn0BUFL in the transmit mode and becomes "0" when the transmit data is transferred to the shift register. To transmit data successively, check that the Un0FUL flag becomes "0", then write the next transmit data in SD0BUFL.
  When "1" is written in Un0FUL, it is reset to "0".
  Un0FUL is fixed to "0" in the receive mode.

Un0FUL	Description	
	Full-duplex communication mode	Half-duplex communication mode
0	No data in the transmit buffer (SD0BUFH)(initial value)	No data in the transmit buffer (SD0BUFL) (initial value)
1	Data present in the transmit buffer (SD0BUFH)	Data present in the transmit buffer (SD0BUFL) during transmit operation

- **Un0TXF** (Bit 4)
  Un0TXF is a bit to indicate that UART is transmitting data in UARTn0 full-duplex communiction mode and half-duplex communication mode.

Un0TXF	Description
0	Data transmission stopped (initial value)
1	Data transmission in progress

- **Un0RXF** (Bit 5)
  Un0RXF is a bit to indicate that UART is receiving data in UARTn0 full-duplex communiction mode and half-duplex communication mode.

Un0RXF	Description
0	Date reception stopped (initial value)
1	Data reception in progress

[Note]
- The Un0OER bit becomes "1" if the previous receive data is not read even after reception is stopped by the Un0EN bit and restarted. Therefore, set the Un0EN bit to “1” after reading the SDnBUFL, or when reception is completed, be sure to read the SDnBUFL even if the data is not necessary.
- When an error occurs in the start bit, the state returns to the reception waiting state.
- Do not write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by using the bit symbol. Write them by the byte-access.

11.2.15 UARTn1 Status Register (UAn1STAT)

Address: 0xF61A, 0xF63A
Access: R/W
Access size: 8 bits
Initial value: 00H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								UAn1STAT							
Bit symbol	-	-	-	-	-	-	-	-	-	-	Un1RXF	Un1TXF	Un1FUL	Un1PER	Un1OER	Un1FER
Access type	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

UAn1STAT is a specific function register (SFR) to indicate the state in the transmit/receive operation in UARTn1 half-duplex communication mode.
When the full-duplex communication mode is selected, the contents of the UAn1STAT register become invalid.

Description of bits

- **Un1FER** (Bit 0)
  Un1FER is a bit to indicate a framing error in UARTn1 half-duplex communication mode.
  When the half-duplex communication mode is selected, the Un1FER becomes "1" if an error occurs in the stop bit and holds “1” until it is cleared by the software.
  When "1" is written in Un1FER, it is reset to "0."
  Un1FER is fixed to "0" in the transmit mode.

Un1FER	Description
0	Framing error has not occurred (initial value)
1	Framing error has occurred

- **Un1OER** (Bit 1)
  Un1OER is a bit to indicate an overrun error in UARTn1 half-duplex communication mode.
  When the half-duplex communication mode is selected, it becomes "1" if data is received again before the receive data in the transmit/receive buffer (UAn1BUF) is read.
  When "1" is written in Un1OER, it is reset to "0."
  Un1OER is fixed to "0" in the transmit mode.

Un1OER	Description
0	Overrun error has not occurred (initial value)
1	Overrun error has occurred

- **Un1PER** (Bit 2)
  Un1PER is a bit to indicate a parity error in UARTn1 half-duplex communication mode.
  The parity of the received data and the parity bit added to the data are compared and if they do not match the Un1PER becomes "1" and holds “1” until it is cleared by the software.
  When "1" is written in Un1PER, it is reset to "0."
  Un1PER is fixed to "0" in the transmit mode.

Un1PER	Description
0	Parity error has not occurred (initial value)
1	Parity error has occurred

- **Un1FUL** (Bit 3)
  Un1FUL is a bit to indicate the state of the transmit/receive buffer in UARTn1 half-duplex communication mode.
  It becomes "1" when transmit data is written in SD0BUFH in the transmit mode and "0" when the transmit data is transferred to the shift register. To transmit data successively, check that the Un1FUL flag becomes "0", then write the next transmit data in SD0BUFH.
  When "1" is written in Un1FUL, it is reset to "0."
  Un1FUL is fixed to "0" in the receive mode.

Un1FUL	Description
0	Transmit/receive buffer has no data (initial value)
1	Transmit/receive buffer has data

- **Un0TXF** (Bit 4)
  Un1TXF is a bit to indicate that UART is transmitting data in UARTn1 half-duplex communication mode.
  It becomes "1" when data to be output to the TXD pin is being transferred by the shift register and "0" when transmission is completed.

Un1TXF	Description
0	Data transmission stopped (initial value)
1	Data transmission in progress

- **Un1RXF** (Bit 5)
  Un1RXF is a bit to indicate that UART1 is receiving data in UARTn1 half-duplex communication mode.
  It becomes "1" when data input from the RXD pin is being transferred by the shift register and "0" when reception is completed.

Un1RXF	Description
0	Date reception stopped (initial value)
1	Data reception in progress

[Note]
- The Un1OER bit becomes "1" if the previous receive data is not read even after reception is stopped by the Un1EN bit and restarted. Therefore, set the Un1EN bit to “1” after reading the SDnBUFH, or when reception is completed, be sure to read the SDnBUFH even if the data is not necessary.
- When an error occurs in the start bit, the state returns to the reception waiting state.
- Do not write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by using the bit symbol. Write them by the byte-access.

11.3 Description of Operation

11.3.1 Synchronous Serial Port (SSIO)

11.3.1.1 Transmit Operation Timing

Figure 11-2 shows the transmit operation waveform (with 8-bit length, LSB first) of the synchronous serial port for clock type 0 (positive logic). Figure 11-3 shows the one for clock type 0 (negative logic), Figure 11-4 the one for clock type 1 (positive logic), and Figure 11-5 the one for clock type 1 (negative logic).

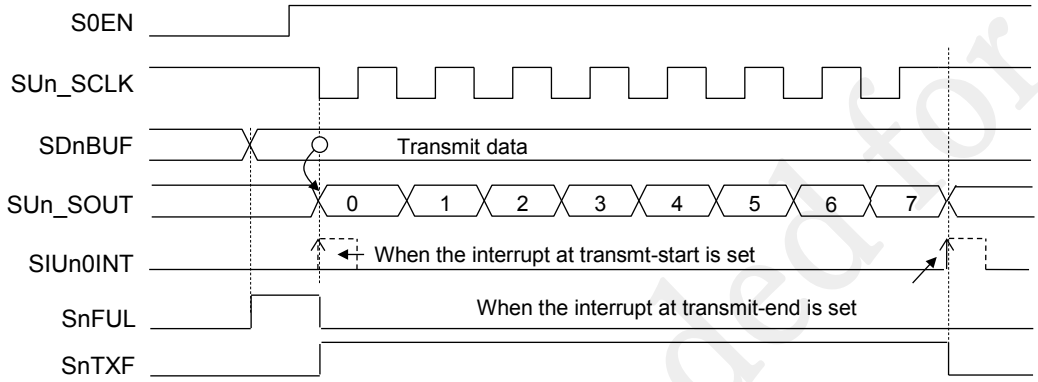


Figure 11-2 Transmit Waveform of SSIO (with Clock Type 0/Positive Logic, 8-Bit Length and LSB First)

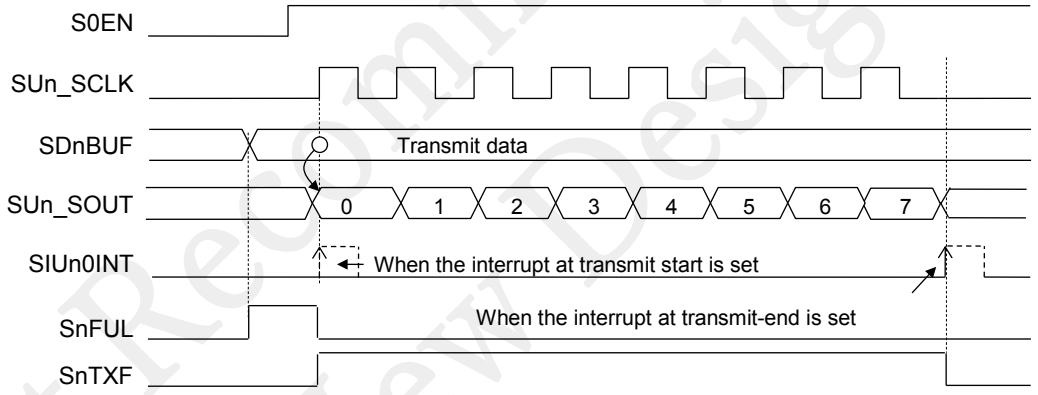


Figure 11-3 Transmit Waveform of SSIO (with Clock Type 0/Negative Logic, 8-Bit Length and LSB First)

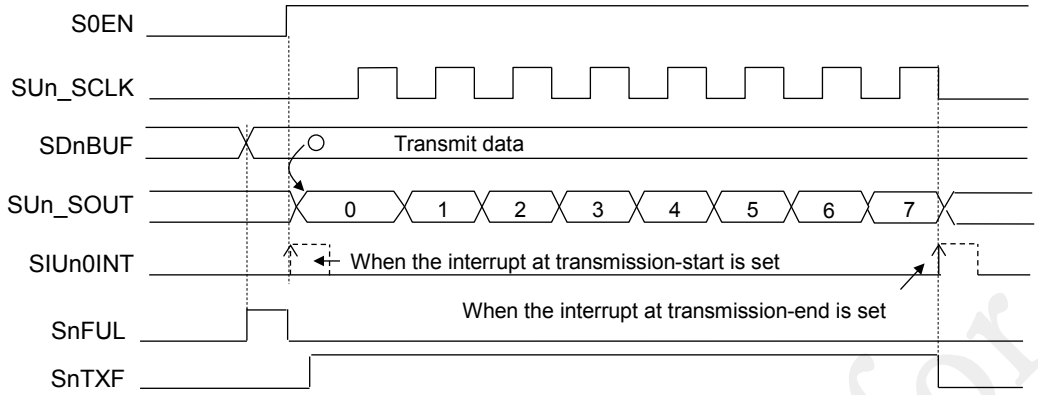


Figure 11-4 Transmit Waveform of SSIO (with Clock Type 1/Positive Logic, 8-Bit Length and LSB First)

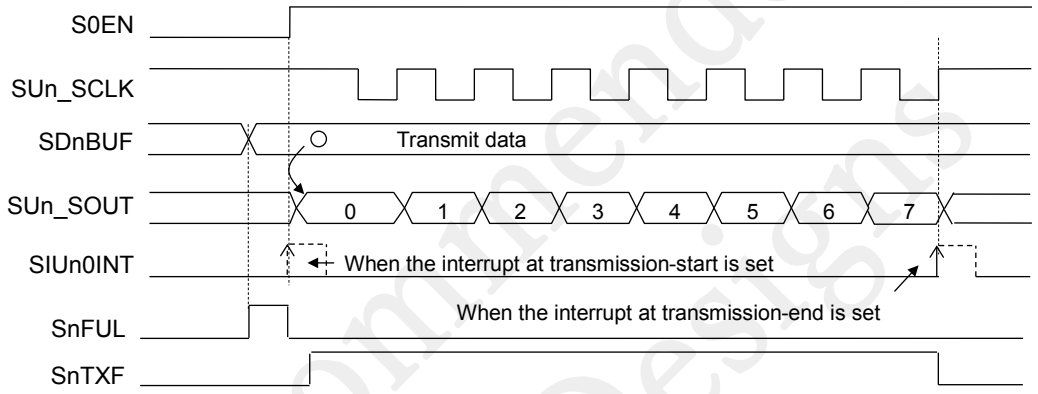


Figure 11-5 Transmit Waveform of SSIO (with Clock Type 1/Negative Logic, 8-Bit Length and LSB First)

11.3.1.2 Receive Operation Timing

Figure 11-6 shows the receive operation waveform (with 8-bit length, MSB first) of the synchronous serial port for clock type 0 (positive logic). Figure 11-7 shows the one for clock type 0 (negative logic), Figure 11-8 the one for clock type 1 (positive logic), and Figure 11-9 the one for clock type 1 (negative logic).
SUn_SCLK is output by writing dummy transmit data into the SDnBUF, means that writing the dummy data is required to received data. Use the transmit/receive mode to have a frame interval when receiving.

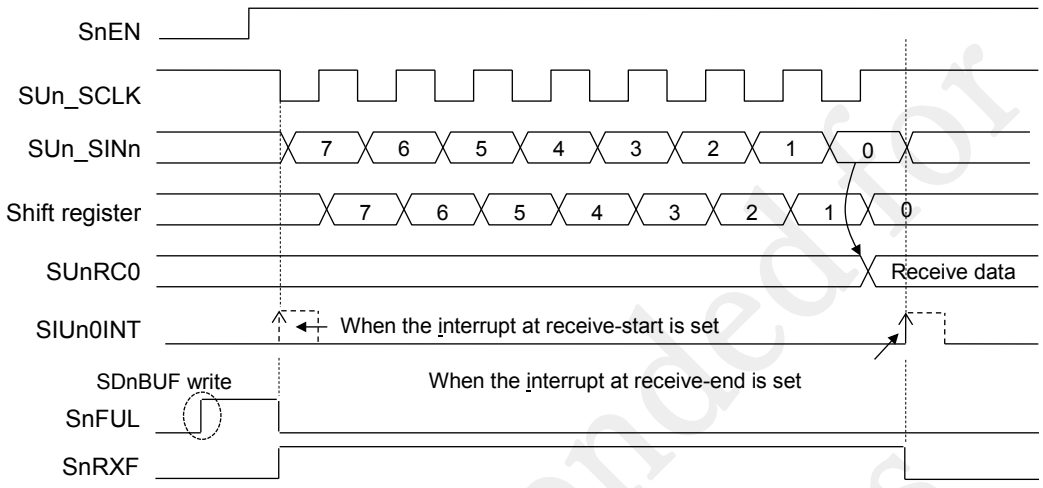


Figure 11-6 Receive Waveform of SSIO (with Clock Type 0/Positive Logic, 8-Bit Length and MSB First)

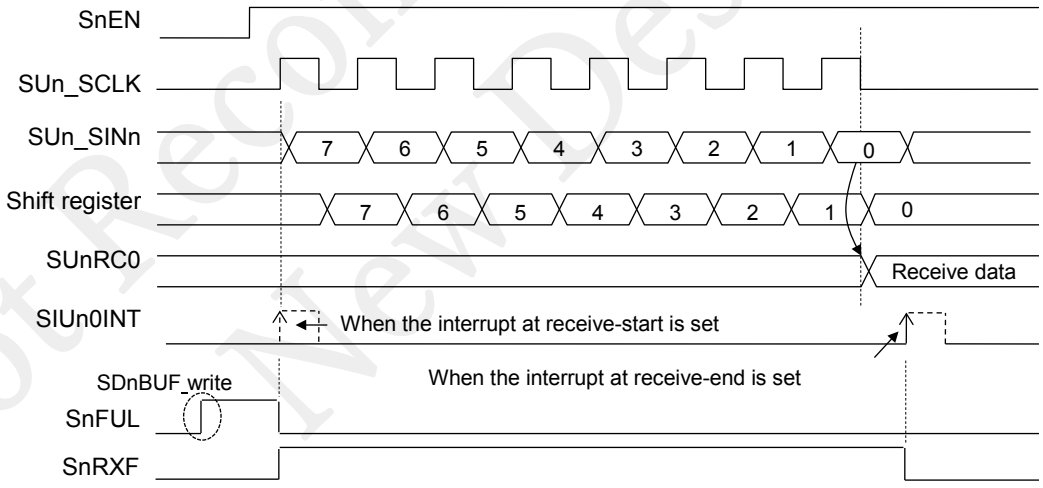


Figure 11-7 Receive Waveform of SSIO (with Clock Type 0/Negative Logic, 8-Bit Length and MSB First)

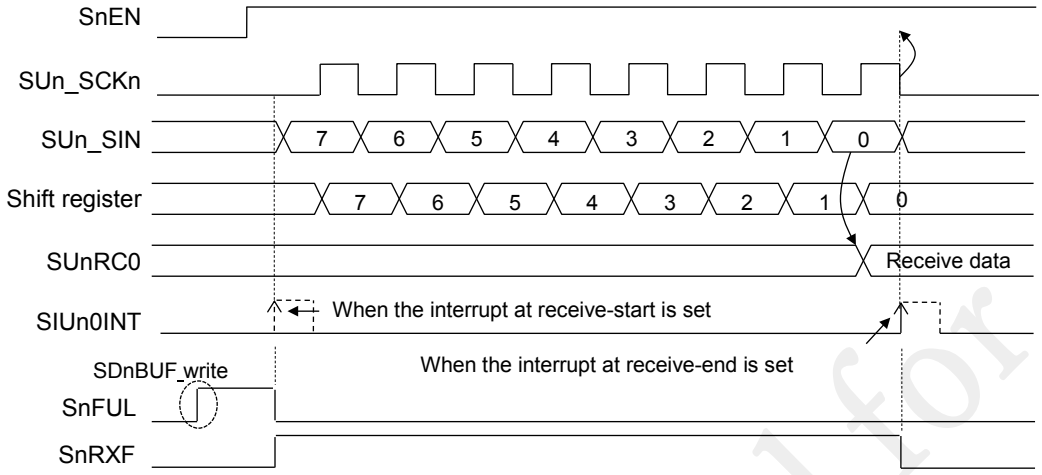


Figure 11-8 Receive Waveform of SSIO (with Clock Type 1/Positive Logic, 8-Bit Length and MSB First)

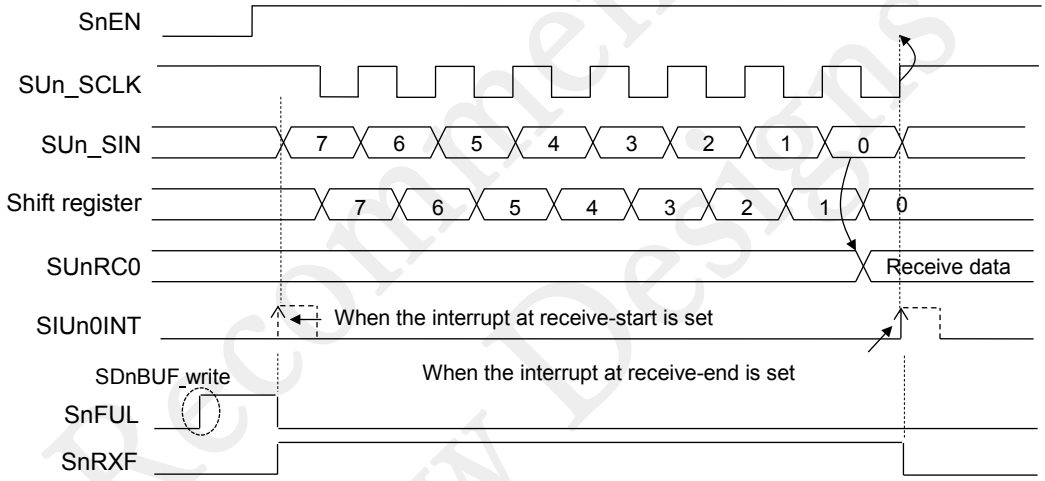


Figure 11-9 Receive Waveform of SSIO (with Clock Type 1/Negative Logic, 8-Bit Length and MSB First)

11.3.1.3 Transmit/Receive Operation Timing

Figure 11-10 shows the transmit/receive operation waveform (with 16-bit length, LSB first, clock type 0) of the synchronous serial port.

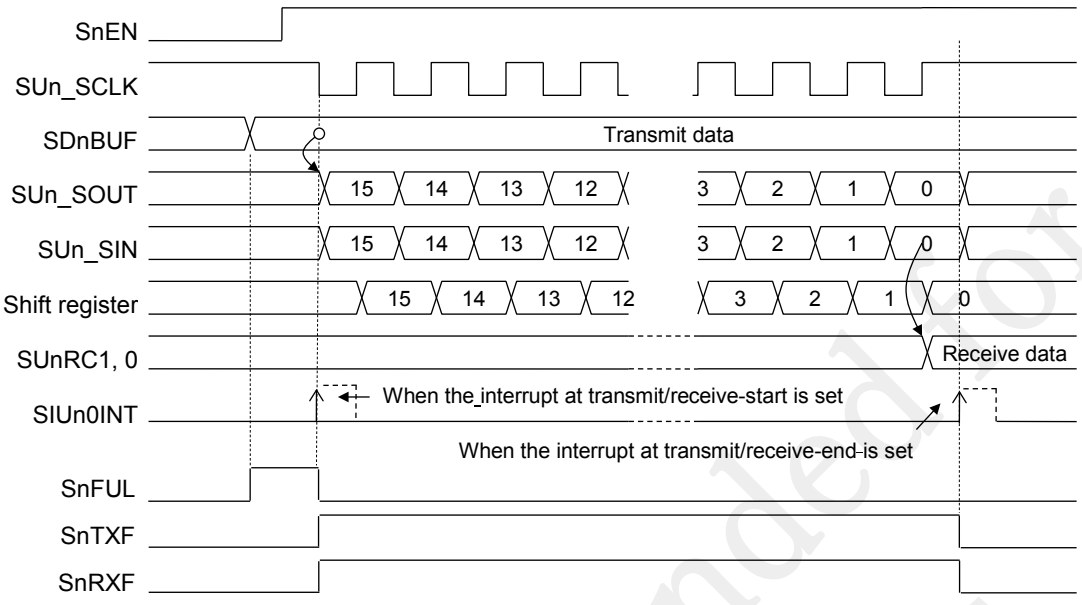


Figure 11-10 Transmit/Receive Waveform of SSIO (with Clock Type 0/Positive Logic, 16-Bit Length and MSB First)

11.3.1.4 Interrupt Timing

The Table 11-2 and 11-3 show the interrupt timing in the transmit mode, receive mode and transmit/receive mode.

Table 11-2 Interrupt timing of transmit mode, receive mode and transmit/receive mode (Master)

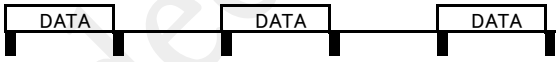
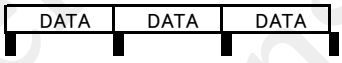
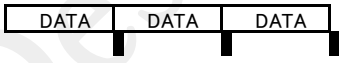
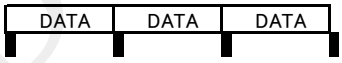
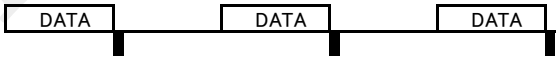
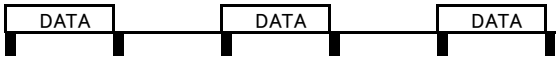
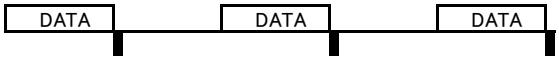
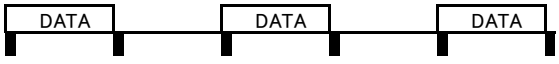
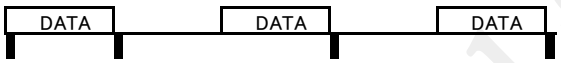
Operation mode	Interrupt occurrence timing	transmission interval setting	SnFUL	Interrupt occurrence timing (▮)
Transmit	Transmit end	No set	1 (The buffer has data)	
	Transmit start and end	No set	1 (The buffer has data)	
	Transmit end	Set	1 (The buffer has data)	
	Transmit start and end	Set	1 (The buffer has data)	
	Transmit end	–	0 (The buffer has no data)	
	Transmit start and end	–	0 (The buffer has no data)	
Receive	Receive end	–	1 (The buffer has data)	
	Receive start and end	–	1 (The buffer has data)	
	Receive end	–	0 (The buffer has no data)	
	Receive start and end	–	0 (The buffer has no data)	
Transmit/ Receive	Transmit/Receive end	No set	1 (The buffer has data)	
	Transmit/Receive start and end	No set	1 (The buffer has data)	
	Transmit/Receive end	Set	1 (The buffer has data)	
	Transmit/Receive start and end	Set	1 (The buffer has data)	
	Transmit/Receive end	–	0 (The buffer has no data)	
	Transmit/Receive start and end	–	0 (The buffer has no data)	

Table 11-3 Interrupt timing of transmit mode, receive mode and transmit/receive mode (Slave)

Operation mode	Interrupt occurrence timing	Interrupt occurrence timing (▮)
Transmit	Transmit end	
	Transmit start and end	
Receive	Receive end	
	Receive start and end	
Transmit/ Receive	Transmit/Receive end	
	Transmit/Receive start and end	

11.3.1.5 Flow chart for the data transfer

The Figure 11-11 shows the flow chart.

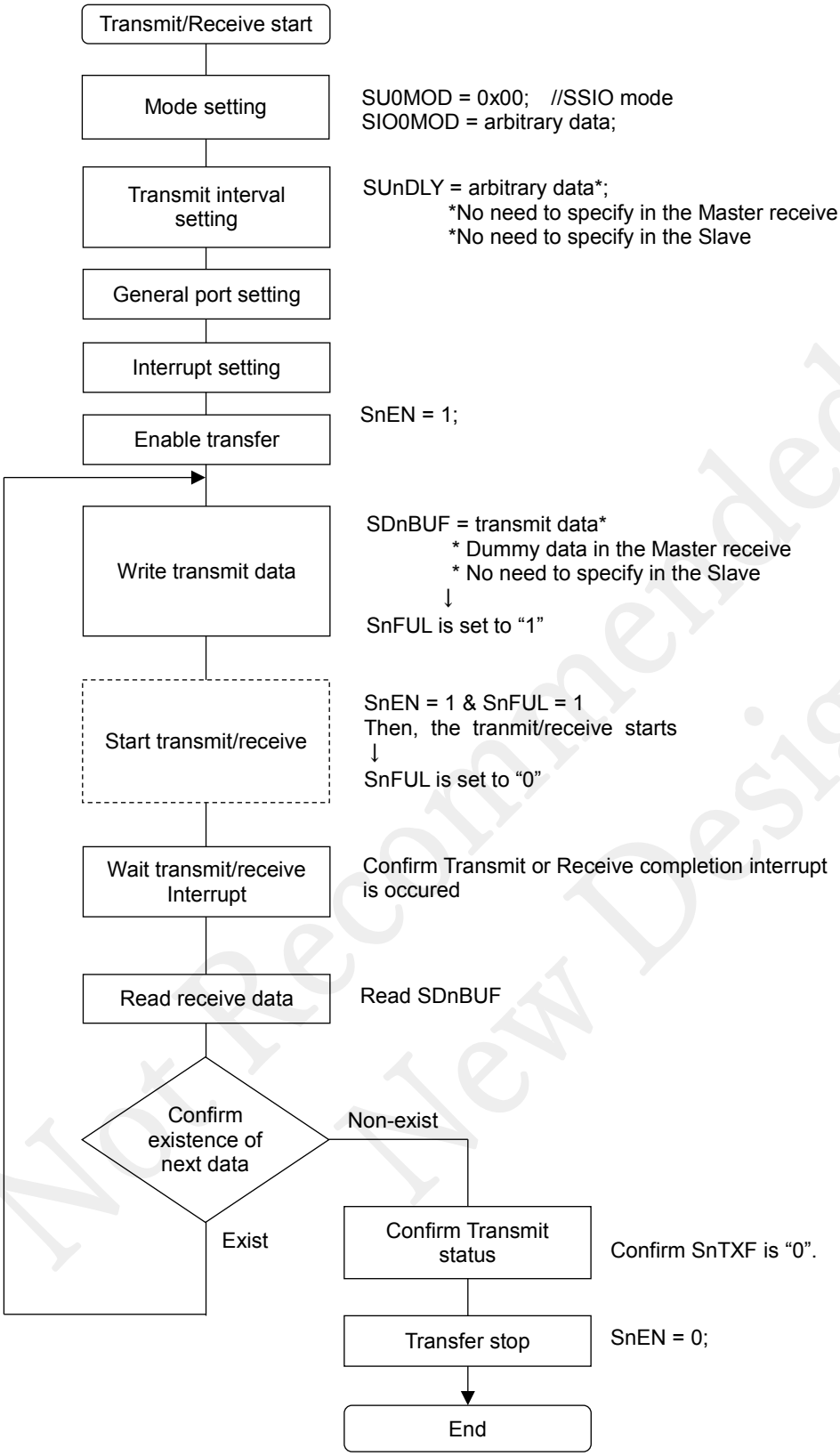


Figure 11-11 Flow Chart of SSIO transfer

11.3.2 Asynchronous Serial Interface (UART)

11.3.2.1 Transfer Data 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 selectable: 5 to 8 bits are selectable for the data bit, availability of parity, even/odd parity, parity fixed to "1", and parity fixed to "0" for the parity bit, 1 stop bit and 2 stop bit for the stop bit, LSB first and MSB first for the transfer direction, and positive logic and negative logic for the logic of the serial input/output.
All of these are set in the UARTn mode register (UAnMOD).
Figure 11-11 and Figure 11-12 show the positive logic input/output format and negative logic input/output format, respectively.

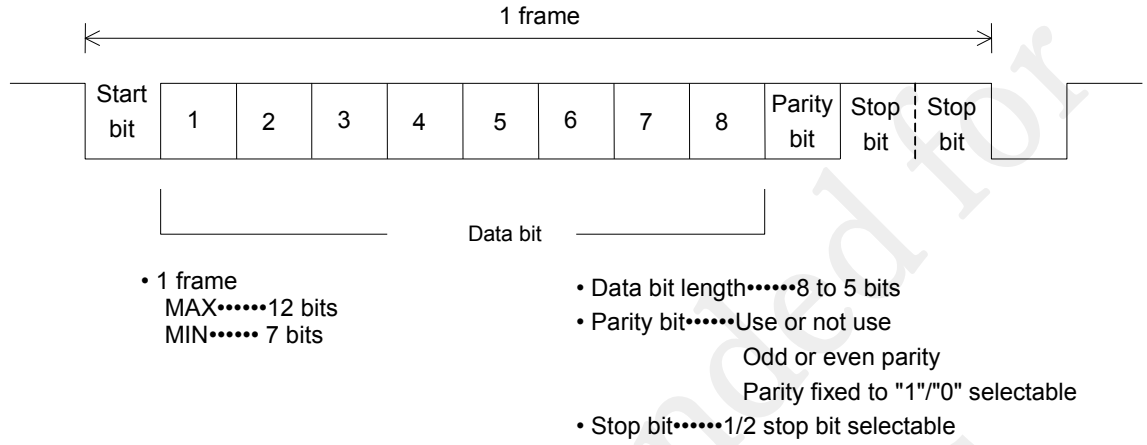


Figure 11-11 Format of Positive Logic Input/Output (LSB First)

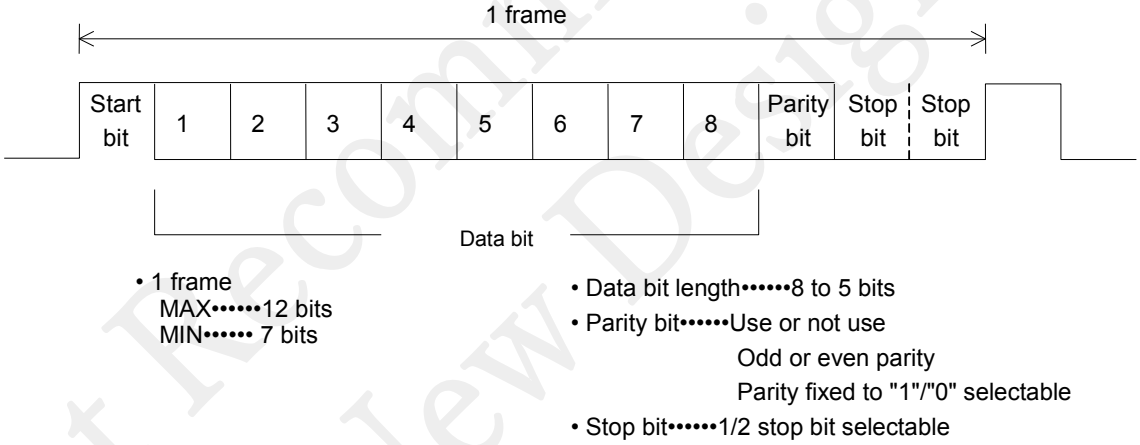


Figure 11-12 Format of Negative Logic Input/Output (LSB First)

11.3.2.2 Baud Rate

The baud rate generator generates a baud rate using the base clcok selected by the UARTn mode register (UAnMOD). The setting values for the UARTn baud rate register (UAn0BRT and UAn1BRT) and UARTn baud rate adjustment register (UAn0BRC and UAn1BRC) can be specified by the following formula.

UAn0BRT = Base clock frequency (Hz) / Baud rate (bps) - 1
UAn0BRC = (Base clock frequency (Hz) %[*1] Baud rate (bps)) x 8 / Baud rate (bps)

When the CPU operation clock is set to 24 MHz, the setting values are usually calculated assuming that the base clock frequency for calculation of the UART baud rate is 23.986176 MHz (15.990784 MHz when 16 MHz). This value is "the central value set for built-in oscillation + PLL oscillation". An error of each LSI unit can be minimized by measuring the frequency for each LSI unit and adjusting it to align with the central value.

For example: Base clock frequency is approx.24MHz (23.986176MHz), Baud rate is 115,200bps
UAn0BRT = 23.986176MHz / 115,200bps - 1
= 208.21333⋯ - 1 = 207 (rounding down to the nearest integar)
= 0x00CF

UAn0BRC = (23.986176MHz % 115,200bps) x 8 / 115,200bps
=(24576 x 8) / 115,200
= 196608 / 115,200
= 1.70666⋯ = 2 (rounding to the nearest integar)
= 0x02

The actual baud rate is calculated from the setting values by the following formula.
Actual baud rate (bps) = [Base clock] / { (UA0BRT + 1) + (UAn0BRC / 8) }

For example: Base clock frequency is approx.24MHz (23.986176MHz), Baud rate is 1,200bps
Actual baud rate (bps) = 23.986176MHz / { (0x4E13 + 1) + (0x04376) }
= 1199.99

Table 11-4 shows examples of the setting values for common baud rates.

Table 11-4 Examples of the setting values for common baud rates

Base clock	Ideal Baud Rate	UAn0BRT UAn1BRT	UAn0BRC UAn1BRC	Actual Baud Rate
Approx. 32MHz (31.981568MHz)	1,200bps	0x681A	0x02	1200.00bps
	2,400bps	0x340C	0x05	2400.00bps
	4,800bps	0x1A05	0x07	4799.96bps
	9,600bps	0x0D02	0x03	9600.11bps
	19,200bps	0x0680	0x06	19199.50bps
	38,400bps	0x033F	0x07	38399.00bps
	57,600bps	0x022A	0x02	57598.50bps
	115,200bps	0x0114	0x05	115197.00bps
Approx.24MHz (23.986176MHz)	1,200bps	0x4E13	0x04	1199.99bps
	2,400bps	0x2709	0x02	2399.99bps
	4,800bps	0x1384	0x01	4799.99bps
	9,600bps	0x09C1	0x04	9600.23bps
	19,200bps	0x04E0	0x02	19200.46bps
	38,400bps	0x026F	0x05	38400.92bps
	57,600bps	0x019F	0x03	57607.14bps
	115,200bps	0x00CF	0x02	115179.71bps
Approx.16MHz (15.990784MHz)	300bps	0xD035	0x05	299.99bps
	1,200bps	0x340C	0x05	1200.00bps
	2,400bps	0x1A05	0x07	2399.98bps
	4,800bps	0x0D02	0x03	4800.05bps
	9,600bps	0x0680	0x06	9599.75bps
	19,200bps	0x033F	0x07	19199.50bps
	38,400bps	0x019F	0x03	38404.76bps
	57,600bps	0x0114	0x05	57598.50bps
	115,200bps	0x0089	0x06	115248.89bps
Approx.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

11.3.2.3 Transmitted Data Direction

Figure 11-13 shows the relationship between the transmit/receive buffer and transmit/receive data.

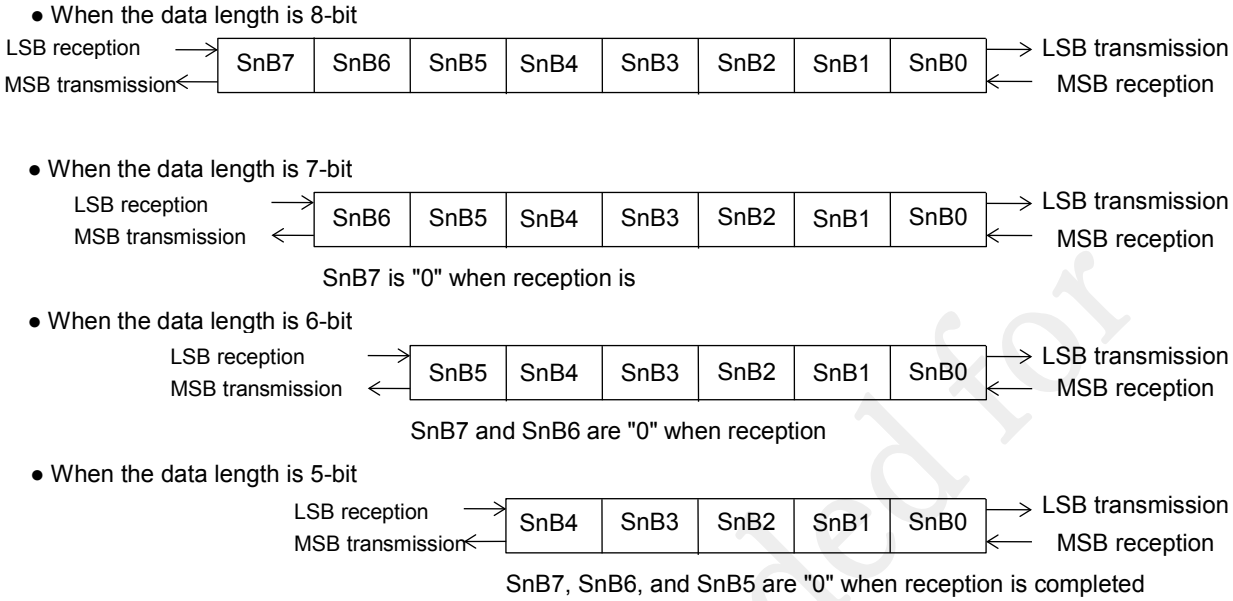


Figure 11-13 Relationship between Transmit/Receive Buffer and Transmit/Receive Data

[Note]
When the SUn_TXDn pin is set to non-primary function in receive mode, "H" level is output from the SUn_TXDn pin.

11.3.2.4 Transmit Operation

Refer to the following transmit procedure in the UART full-duplex communication mode. Figure 11-14 shows the operation timing for transmit.

- Preparing the communication (transmit/receive common settings for full-duplex communication mode)
  - Select the full-duplex communication mode by the Serial Communication Unit n Mode Register (SUnMOD)
  - Set the Serial Communication Unit n Transmission Interval Setting Register (SUnDLYL) if using the function.
  - Select the communication mode by the UARTn0 Mode Register (UAn0MOD).
  - Set the baud rate by UARTn0 Baud Rate Register (UAn0BRT) and UARTn0 Baud Rate Adjustment Register (UAn0BRC).
  - Select UART communication pins among assigned genearl ports.
  - Clear the serial communication unit n0 and n1 interrupts (QSIUn0=0, QSIUn1=0).
  - Enable the serial communication unit n0 and n1 interrupts (ESIUn0=1, ESIUn1=1).
  - Read out the serial communication unit n transmid/receive bufffer L (SDnBUFL) to prevent miss-detection of overrun errors. No need the read data.
  - Write "0xFF" into the UARTn0 Statsu Register (UAn0STAT) to clear each flags.
- Starting the transmit
  - Enable transmit/receive by setting Un0EN bit of Serial Communication Unit n Control Register L (SUnCONL) (1)
  - Un0FUL bit is set to "1" by writing the transmit data in the Serial Communication Unit n Transmit/Receive Buffer H (SDnBUFH) (2), the baud rate generator generates the internal transfer clock and transmit starts. After starting the transmit, the start bit is output to the Sun_TXD1 at the falling edge of the internal transfer clock and Un0FUL gets to "0" (3).
  - At this time, the Serial Communication Unit n1 Interrupt (SIUn1INT) occurs if the interrupt is selected by the Serial Communication Unit n Mode Register (SUnMOD).
- Successive transmit
  - Write the next transmit data in the SDnBUFH after checking the Serial Communication Unit n1 Interrupt or checking the Un0FUL bit is "0" (4). The Un0FUL bit is set to "1" after writing the data. The successsive transmit is available by writing the next transmit data during the "transmit/receive buffer write valid period"(5).
  - The Serial Communication Unit n1 Interrupt (SIUn1INT) occurs after the stop bit of the first data is transmitted (6). If the Un0FUL bit is "1", it continue to tramsmit the next data.
- Ending the transmit
  - If the last transmit is completed (no new data has been written in the SDnBUFH) the successive transmit ends and the Serial Communication Unit n1 Interrupt occurs (SIUn1INT) (7).
  - Continue to write transmit data in the SDnBUFH if need to make the successive transmit. Clear the Un0EN bit of the SUnCONL to end the all UART transmit/receive.

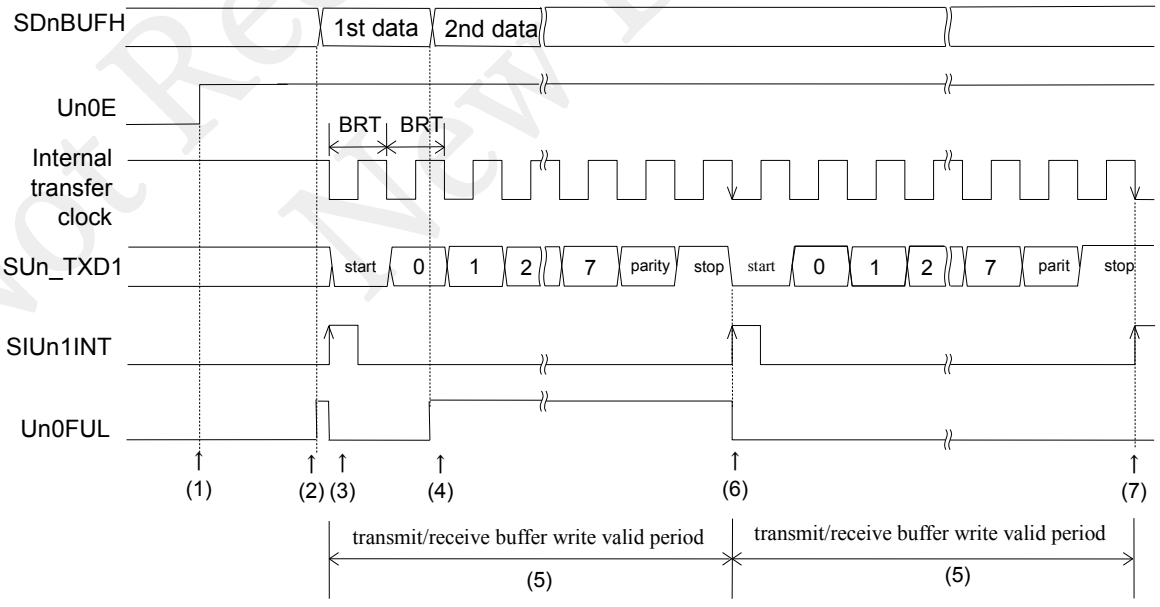


Figure 11-14 Operation Timing in Transmission

11.3.2.5 Receive Operation

Refer to the following receive procedure in the UART full-duplex communication mode. Figure 11-15 shows the operation timing for receive.

- Preparing the communication (transmit/receive common settings for full-duplex communication mode)
  - Select the full-duplex communication mode by the Serial Communication Unit n Mode Register (SUnMOD)
  - Set the Serial Communication Unit n Transmission Interval Setting Register (SUnDLYL) if using the function.
  - Select the communication mode by the UARTn0 Mode Register (UAn0MOD).
  - Set the baud rate by UARTn0 Baud Rate Register (UAn0BRT) and UARTn0 Baud Rate Adjustment Register (UAn0BRC).
  - Select UART communication pins among assigned genearl ports.
  - Clear the serial communication unit n0 and n1 interrupts (QSIUn0=0, QSIUn1=0).
  - Enable the serial communication unit n0 and n1 interrupts (ESIUn0=1, ESIUn1=1).
  - Read out the serial communication unit n transmid/receive bufffer L (SDnBUFL) to prevent miss-detection of overrun errors. No need the read data.
  - Write “0xFF” into the UARTn0 Statsu Register (UAn0STAT) to clear each flags.
- Starting the receive
  - Enable transmit/receive by setting Un0EN bit of Serial Communication Unit n Control Register L (SUnCONL) (1)
  - The hardware starts detecting the start bit input from SUn_RXD0 pin.
  - If detecting “L” level from the SUn_RXD pin (2), the baud rate generator starts generating a transfer clock. If receive “H” level in the middle of start bit, it takes as wrong operation and goes back detecting the start bit.
  - If receive “L” level in the middle of start bit, it starts receiving operation to get the input data from SUn_RXD0 pin to the shift register at rising edges of internal transfer clock.
  - After getting a receive data and parity bit, they will be transferred to Serial Communication Unit n transmit/receive buffer (SDnBUFL) (3).
  - The Serial Communication Unit n0 Interrupt (SIUn0INT) occurs in the middle of stop bit (4) and detect the framing error (stop bit error) and parity error. If it got the error, corresponding bits (Un0FER, Un0PER) are set. At the same time, it goes to detect the next start bit.
  - In the ase of successive receive, Un0OER bit of UAnSTAT register is set to “1” (indicates over run error) if the received data in SDnBUFL is overwritten before the CPU read out the previous data.
- Ending the receive
  - Clear the Un0EN bit of the SUnCONL to end the UART receive(5). Do not clear the Un0EN on the way of receive, otherwise the receiving data may be corrupted.

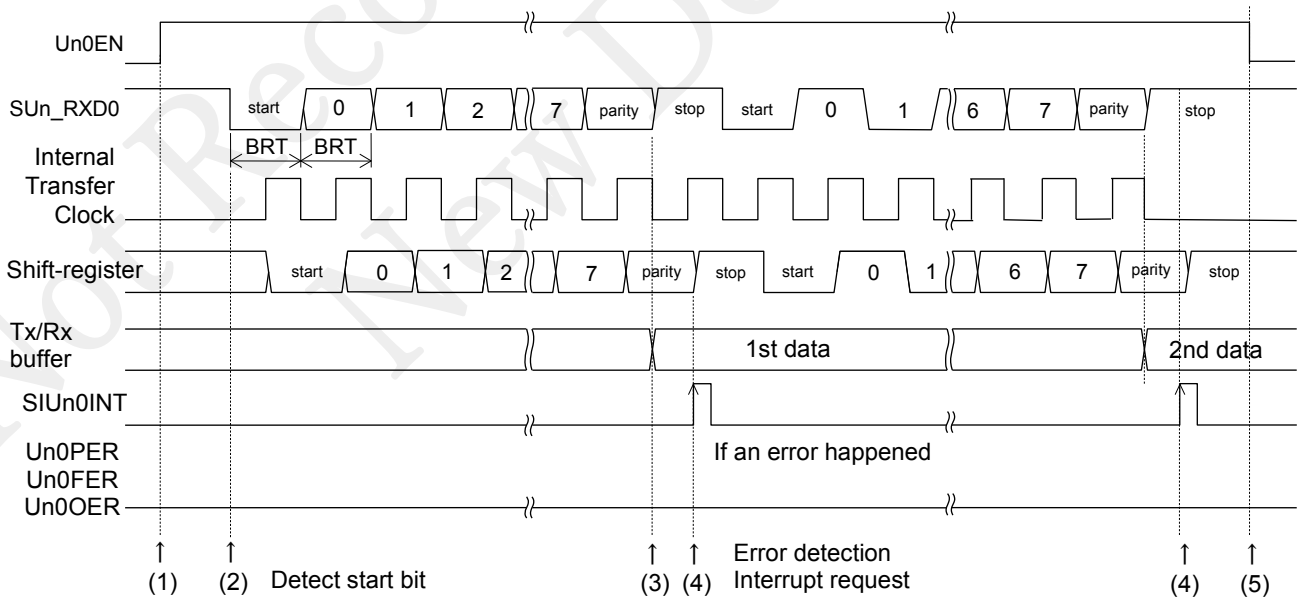


Figure 11-15 Operation Timing in Reception

11.3.2.6 Detection of Start Bit

The start bit is sampled by the base clock of baud rate generator. Therefore, the start bit detection may be delayed for max. one cycle of the baud rate generator clock.
Figure 11-16 shows the start bit detection timing.

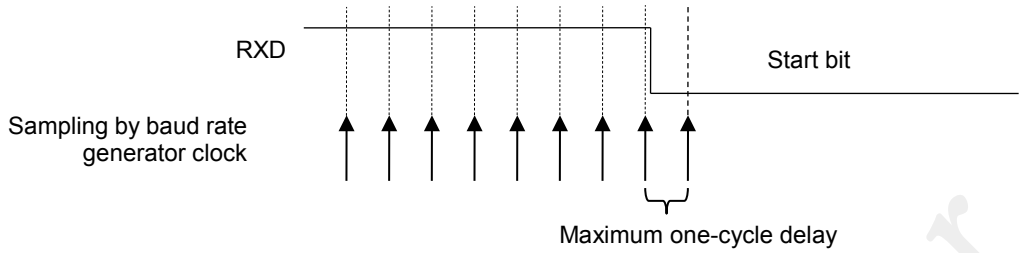
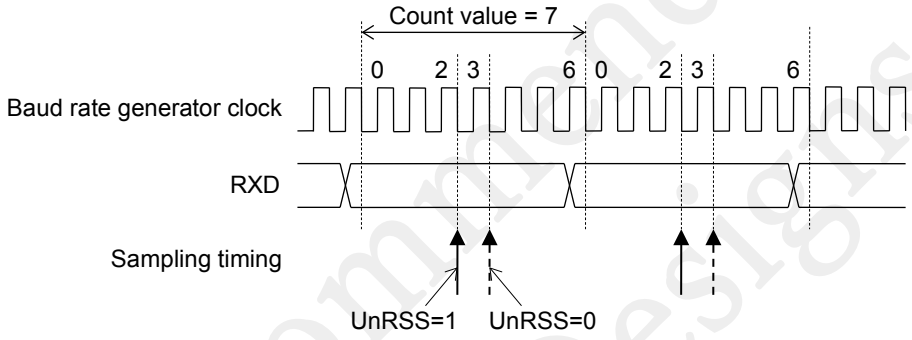


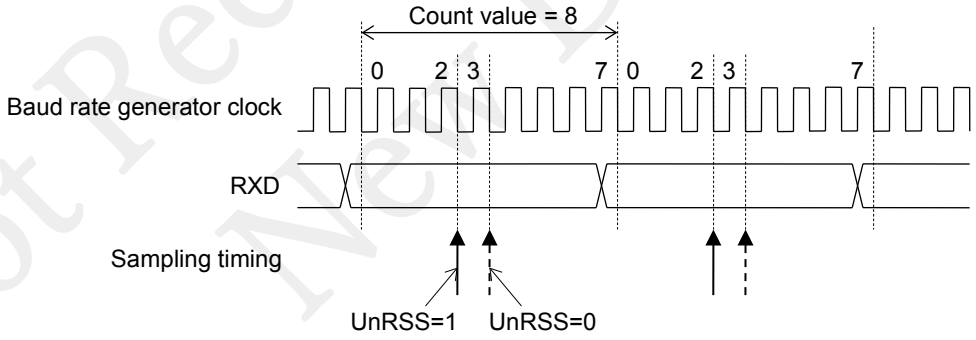
Figure 11-16 Start Bit Detection Timing (with Positive Logic)

11.3.2.7 Sampling Timing

When the start bit is detected, the received data that was input to the Sun_RXDn is sampled almost at the center of the baud rate, then loaded to the shift register.
This sampling timing to capture with the shift register can be adjusted for one clock of the baud rate generator clock in the UnRSS bit of the UARTn mode register 0 (UAn0MODL).
Figure 11-17 shows the relationship between the UnRSS bit and sampling timing.



(1) When the baud rate generator counter value is "7" (odd number)



(2) When the baud rate generator count value is "8" (even value)

Figure 11-17 Relationship between UnRSS Bit and Sampling Timing

11.3.2.8 Receive Margin

If there is an error between the sender baud rate and the baud rate generated by the baud rate generator of this LSI, the error accumulates until the last stop bit loading in one frame, decreasing the receive margin.
Figure 11-18 shows the baud rate errors and receive margin waveforms.

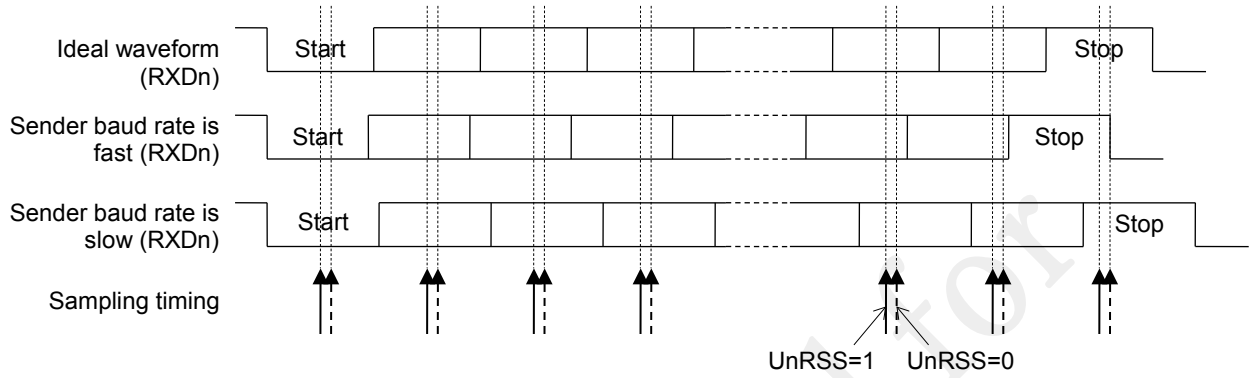


Figure 11-18 Baud Rate Errors and Receive Margin

[Note]
When designing the system, adjust the baud rate in the UAn0BRT, UAn1BRT, UAn0BRC, and UAn1BRC registers, considering difference of the baud rate between the transmit side and receive side, and also the delay of start bit detection.

11.3.2.9 Interrupt Timing

Table 11-5 and 11-6 show the interrupt occurrence timing in the UART transmit and receive.

Table 11-5 Interrupt timing of UART transmit

Interrupt occurrence timing	Transmit interval / S0FUL	Interrupt occurrence timing (■)
Transmit end	Non interval / data exists	DATA DATA DATA
	Non interval / No data With interval / data exists With interval / No data	DATA DATA DATA
Transmit start and end	Non interval / data exists	DATA DATA DATA
	With interval / data exists	DATA DATA DATA
	Non interval / No data With interval / No data	DATA DATA DATA

Table 11-6 Interrupt timing of UART receive

Interrupt occurrence timing	Interrupt occurrence timing (▬)
Receive end	DATA DATA DATA
Receive start and end	DATA DATA DATA

11.3.3 Pin Settings

SUn_SOUT, SUn_SIN and SUn_SCLK can be selected from multiple GPIOs.
When selecting the pins, be sure to use them in this following combinations.

Input/output pin	Combination 1	Combination 2	Combination 3	Combination 4
SU0_SIN	P02 /SU0_SIN	P12/SU0_SIN	P21/SU1_SIN	P24/SU1_SIN
SU0_SOUT	P03 /SU0_SOUT	P13 /SU0_SOUT	P22 /SU1_SOUT	P25 /SU1_SOUT
SCLK	P04 /SU0_SCLK	P11/SU0_SCLK	P16/SU1_SCLK	P23 /SU1_SCLK

Chapter 12 I^2C Bus Unit

12. I^2C Bus Unit

12.1 General Description

ML62Q1000 series has one channel of I^2C bus unit that supports both master and slave function. Either of master or slave is selected to use and both functions cannot work at the same time.

12.1.1 Features

- Supports both master and slave functions
- Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1 Mbps)
- Supports clock synchronization function (Master) and clock stretch function (Slave)
- 7-bit address format (only the master function supports 10-bit address format)
- Self-test function (safety function)

12.1.2 Configuration

Figure 12-1 shows the configuration diagram of the I²C bus unit circuit.

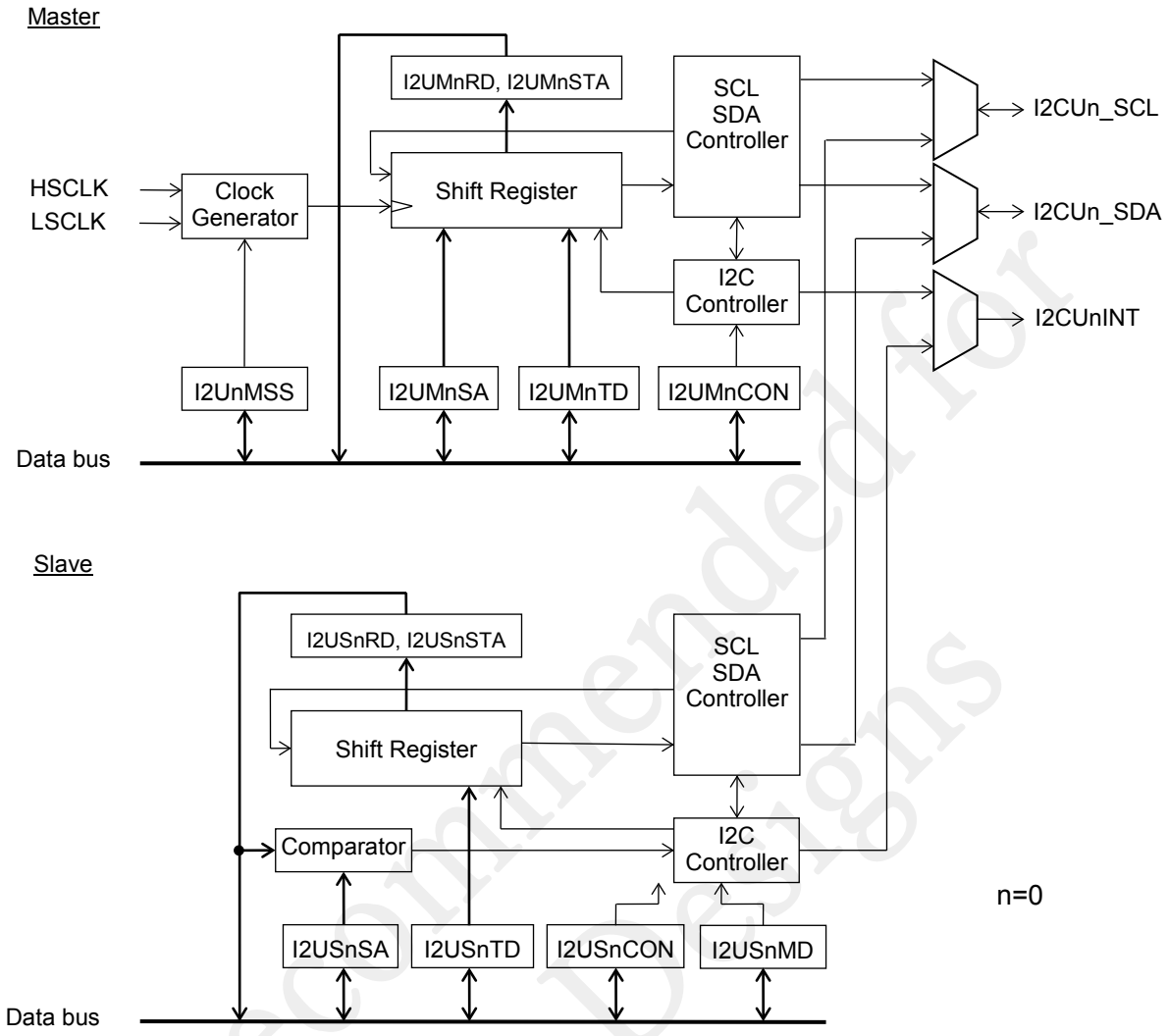


Figure 12-1 Configuration Diagram of I²C Bus Unit

I2UnMSS	: I²C bus unit n mode register
I2UMnRD	: I²C bus n receive register (master)
I2UMnSA	: I²C bus n slave address register (master)
I2UMnTD	: I²C bus n transmit data register (master)
I2UMnCON	: I²C bus n control register (master)
I2UMnMOD	: I²C bus n mode register (master)
I2UMnSTA	: I²C bus n status register (master)
I2USnRD	: I²C bus n receive register (slave)
I2USnSA	: I²C bus n slave address register (slave)
I2USnTD	: I²C bus n transmit data register (slave)
I2USnCON	: I²C bus n control register (slave)
I2USnMD	: I²C bus n mode register (slave)
I2USnSTA	: I²C bus n status register (slave)

12.1.3 List of Pins

I/Os of the I2C bus unit are assigned to the secondary to octic functions of the GPIO.
For details about pin assignment and how to set the secondary to octic functions of the GPIO, see Chapter 17 "GPIO".

Pin name	I/O	Function
I2CUn_SDA	I/O	I^2C bus unit data I/O pin
I2CUn_SCL	I/O	I^2C bus unit clock I/O pin

12.2 Description of Registers

12.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF6C0	I^2C bus unit 0 master/slave select	I2U0MSS	-	R/W	8	0x00
0xF6C1	Reserved register	-	-	R	8	0x00
0xF6C2	I2C bus unit 0 receive register (master)	I2UM0RD	-	R	8	0x00
0xF6C3	Reserved register	-	-	R	8	0x00
0xF6C4	I2C bus 0 slave address register (master)	I2UM0SA	-	R/W	8	0x00
0xF6C5	Reserved register	-	-	R	8	0x00
0xF6C6	I2C bus 0 transmit data register (master)	I2UM0TD	-	R/W	8	0x00
0xF6C7	Reserved register	-	-	R	8	0x00
0xF6C8	I2C bus 0 control register (master)	I2UM0CON	-	R/W	8	0x00
0xF6C9	Reserved register	-	-	R	8	0x00
0xF6CA	I2C bus 0 mode register (master)	I2UM0MDL	I2UM0MOD	R/W	8/16	0x00
0xF6CB		I2UM0MDH		R/W	8	0x02
0xF6CC	I2C bus 0 status register (master)	I2UM0STA	-	R/W	8	0x00
0xF6CD	Reserved register	-	-	R	8	0x00
0xF6CE	I2C bus 0 receive register (slave)	I2US0RD	-	R	8	0x00
0xF6CF	Reserved register	-	-	R	8	0x00
0xF6D0	I2C bus 0 slave address register (slave)	I2US0SA	-	R/W	8	0x00
0xF6D1	Reserved register	-	-	R	8	0x00
0xF6D2	I2C bus 0 transmit data register (slave)	I2US0TD	-	R/W	8	0x00
0xF6D3	Reserved register	-	-	R	8	0x00
0xF6D4	I2C bus 0 control register (slave)	I2US0CON	-	R/W	8	0x00
0xF6D5	Reserved register	-	-	R	8	0x00
0xF6D6	I2C bus 0 mode register (slave)	I2US0MD	-	R/W	8	0x00
0xF6D7	Reserved register	-	-	R	8	0x00
0xF6D8	I2C bus 0 status register (slave)	I2US0STA	-	R/W	8	0x00
0xF6D9	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

12.2.2 I²C Bus Unit 0 Mode Register (I2U0MSS)

Address: 0xF6C0
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2U0MSS							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2U0MD
Access type	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

I2U0MD is a special function register (SFR) used to select the Master mode or Slave mode of the I²C bus unit.

Description of bits

- **I2U0MD** (bit 0)
  I2U0MD is a bit used to select the Master mode or Slave mode of the I²C but unit.

I2U0MD	Description
0	Master mode (initial value)
1	Slave mode

[Note]
- Do not write to SFRs for slave function in the master mode and do not write SFRs for master function in the slave mode.
- If powering off this LSI in the slave mode, it disables communications of other devices on the I2C bus. Remain the power to this LSI when it works as a slave mode until the master device is powered off.
- When using the master function, do not connect multiple master devices on the I2C bus.
- When using the salve function, switch the system clock to the high-speed clock if releasing the communication wait status.

12.2.3 I²C Bus 0 Receive Register (Master) (I2UM0RD)

Address: 0xF6C2
Access: R
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2UM0RD							
Bit symbol	-	-	-	-	-	-	-	-	I2UM0R7	I2UM0R6	I2UM0R5	I2UM0R4	I2UM0R3	I2UM0R2	I2UM0R1	I2UM0R0
Access type	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

I2UM0RD is a read-only special function register (SFR) used to store the received data when the Master mode is selected.
I2UM0RD is updated after completion of each reception.

Description of bits

- **I2UM0R7 to I2UM0R0** (bits 7-0)
  The I2UM0R7 to I2UM0R0 bits are used to store the received data when the Master mode is selected. The signal input to the SDA pin is received at transmission of a slave address and at data transmission/reception in sync with the rising edge of the signal on the SCL pin. Since data that has been output to the SDA and SCL pins is received not only at data reception but also at slave address data transmission and data transmission, it can be checked whether transmit data has certainly been transmitted.

12.2.4 I²C Bus 0 Slave Address Register (Master) (I2UM0SA)

Address: 0xF6C4
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2UM0SA							
Bit symbol	-	-	-	-	-	-	-	-	I2UM0A6	I2UM0A5	I2UM0A4	I2UM0A3	I2UM0A2	I2UM0A1	I2UM0A0	I2UM0RW
Access type	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

I2UM0SA is a special function register (SFR) to set the address and transmit/receive mode of the slave device when the Master mode is selected.

Description of bits

- **I2UM0RW** (bit 0)
  The I2UM0RW bit is used to select the data transmit or data receive mode when the Master mode is selected.

I2UM0RW	Description
0	Data transmit mode (initial value)
1	Data Receive Mode

- **I2UM0A6 to I2UM0A0** (bits 7-1)
  The I2UM0A6 to I2UM0A0 bits are used to set the address of the communication partner when the Master mode is selected.

12.2.5 I²C Bus 0 Transmit Data Register (Master) (I2UM0TD)

Address: 0xF6C6
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2UM0TD							
Bit symbol	-	-	-	-	-	-	-	-	I2UM0T7	I2UM0T6	I2UM0T5	I2UM0T4	I2UM0T3	I2UM0T2	I2UM0T1	I2UM0T0
Access type	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

I2UM0TD is a special function register (SFR) used to set the transmit data when the Master mode is selected.

Description of bits

- **I2UM0T7 to I2UM0T0** (bits 7-0)
  The I2UM0T7 to I2UM0T0 bits are used to set the transmit data when the Master mode is selected.

12.2.6 I²C Bus 0 Control Register (Master) (I2UM0CON)

Address: 0xF6C8
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2UM0CON							
Bit symbol	-	-	-	-	-	-	-	-	I2UM0ACT	-	-	-	-	I2UM0RS	I2UM0SP	I2UM0ST
Access type	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

I2UM0CON is a special function register (SFR) used to control transmit and receive operations when the Master mode is selected.

Description of bits

- **I2UM0ST** (bit 0)
  The I2UM0ST bit is used to control the communication operation of the I²C bus unit when the Master mode is selected. When "1" is written to the I2UM0ST bit, communication is started. When "1" is overwritten to the I2UM0ST bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, data transmission/reception restarts. When "0" is written to the I2UM0ST bit, communication is stopped forcibly.
  "1" can be written to the I2UM0ST bit only when the I²C bus unit is in an operation enable state (I2UM0EN = "1").
  When "1" is written to the I2UM0SP bit, the I2UM0ST bit is reset to "0".

I2UM0ST	Description
0	Stops communication (initial value)
1	Starts communication

- **I2UM0SP** (bit 1)
  The I2UM0SP bit is a write-only bit used to request a stop condition when the Master mode is selected. When "1" is written to the I2UM0SP bit, the LSI shifts to the stop condition and communication stops. When the I2UM0SP bit is read, "0" is always read.
  When "1" is written to the I2UM0SP bit, the I2UM0ST bit is reset to "0".

I2UM0SP	Description
0	No stop condition request (initial value)
1	Stop condition request

- **I2UM0RS** (bit 2)
  The I2UM0RS bit is a write-only bit used to request a restart when the Master mode is selected. When "1" is written to this bit during data communication, the LSI shifts to the restart condition and communication restarts from the slave address. "1" can be written to I2UM0RS only while communication is active (I2UM0ST = "1"). When the I2UM0RS bit is read, "0" is always read.

I2UM0RS	Description
0	No restart request (initial value)
1	Restart request

- **I2UM0ACT** (bit 7)
  The I2UM0ACT bit is used to set the acknowledgment data to be output at completion of reception when the Master mode is selected.

I2UM0ACT	Description
0	Acknowledgment data "0" (initial value)
1	Acknowledgment data "1"

[Note]
- Do not update the I2UM0ACT bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.
- When the I2UM0ST bit is “1”, write the I2UM0CON register in the control register setting wait state.

12.2.7 I²C Bus 0 Mode Register (Master) (I2UM0MOD)

Address: 0xF6CA
Access: R/W
Access size: 8/16 bits
Initial value: 0x0200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	I2UM0MOD															
Byte symbol	I2UM0MDH								I2UM0MDL							
Bit symbol	-	-	-	-	-	I2UM0CD2	I2UM0CD1	I2UM0CD0	-	-	I2UM0SYN	I2UM0DW1	I2UM0DW0	I2UM0MD1	I2UM0MD0	I2UM0EN
Access type	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

I2UM0MOD is a special function register (SFR) used to set the operation mode when the Master mode is selected.

Description of bits

- **I2UM0EN** (bit 0)
  The I2UM0EN bit is used to enable the master operation. When "1" is written to the I2UM0EN bit, the I2UM0ST bit can be set and the I2UM0BB flag starts operation. When "0" is written to the I2UM0EN bit, the I²C master stops operation and the I2UM0RD, I2UM0SA, I2UM0TD, and I2UM0CON registers are initialized.

I2UM0EN	Description
0	Stops I²C master operation (initial value)
1	Enables I²C master operation

- **I2UM0MD1 to I2UM0MD0** (bits 2-1)
  The I2UM0MD1 to I2UM0MD0 bits are used to set the communication speed of the I²C bus unit when the Master mode is selected. The standard mode, fast mode, or high speed mode can be selected.

I2UM0MD1	I2UM0MD0	Description
0	0	Standard mode (initial value)/100 kbps*
0	1	Fast mode/400 kbps*
1	0	High speed mode/1 Mbps*
1	1	

* When the I2UM0CD[2:0] is “000” and I2UM0SYN bits is “0”

- **I2UM0DW1 to I2UM0DW0** (bits 4-3)
The I2UM0DW1 to I2UM0DW0 bits are used to set the communication speed reduction rate of the I^2C bus unit when the Master mode is selected. Set this bit so that the communication speed does not exceed 100 kbps/400 kbps/1 Mbps. When LSCLK ~ 1/8 LSCLK is used as the clock (I2UM0CD2 = "1"), "No communication speed reduction" is selected regardless the value of I2UM0DW1 and I2UM0DW0.

I2UM0DW1	I2UM0DW0	Description
0	0	No communication speed reduction (initial value)
0	1	10% communication speed reduction
1	0	20% communication speed reduction
1	1	30% communication speed reduction

- **I2UM0SYN** (bit 5)
The I2UM0SYN bit is used to select whether or not to use the clock synchronization (handshake) function when the Master mode is selected. Set the bit to "1" when using the clock synchronization function. By setting the bit is set to "1", monitors the I^2C bus, threfore, the communication speed gets lower depending on the load of I^2C bus.

I2UM0SYN	Description
0	Disables clock synchronization (initial value)
1	Enables clock synchronization

- **I2UM0CD2 to I2UM0CD0** (bits 10-8)
The I2UM0CD2 to I2UM0CD0 bits are used to set the operating frequency of I^2C when the Master mode is selected.
Table 12-1 and Table 12-2 show the relationship between the setting values of HSCLK, I2UM0CD2, 1, and 0 and the communication speed. Table 12-3 shows the relationship between the setting values of LSCLK, I2UM0CD2, 1, and 0 and the communication speed.

I2UM0CD2	I2UM0CD1	I2UM0CD0	Description
0	0	0	HSCLK
0	0	1	1/2HSCLK
0	1	0	1/4HSCLK (initial value)
0	1	1	Do not use
1	0	0	LSCLK
1	0	1	1/2LSCLK
1	1	0	1/4LSCLK
1	1	1	1/8LSCLK

[Note]
- Specify the I^2C operation clock as follows.
  When the HSCLK is 32MHz
    Standard mode : I^2C operation clock HSCLK~1/4HSCLK
    Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
    1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK
  When the HSCLK is 24MHz
    Standard mode : I^2C operation clock HSCLK~1/4HSCLK
    Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
    1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK
  When the HSCLK is 16MHz
    Standard mode : I^2C operation clock HSCLK, 1/2HSCLK
    Fast mode : I^2C operation clock HSCLK
    1Mbps mode : I^2C operation clock HSCLK

12.2.8 I²C Bus 0 Status Register (Master) (I2UM0STA)

Address: 0xF6CC
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2UM0STA							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	I2UM0ER	I2UM0ACR	I2UM0BB
Access type	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

I2UM0STA is a special function register (SFR) to indicate the state of the I²C bus unit when the Master mode is selected.

Description of bits

- **I2UM0BB** (bit 0)
  The I2UM0BB bit is used to indicate the state of use of the I²C bus when the Master mode is selected. When the start condition is generated on the I²C bus, this bit is set to "1" and when the stop condition is generated, the bit is reset to "0". When "1" is written to the I2UM0BB bit, the I2UM0BB bit is reset to "0".

I2UM0BB	Description
0	I²C bus-free state (initial value)
1	I²C bus-busy state

- **I2UM0ACR** (bit 1)
  The I2US0ACR bit is used to store the acknowledgment signal received when the Master mode is selected. Acknowledgment signals are received each time the slave address is received and data transmission or reception is completed. When "1" is written to the I2UM0ACR bit, the I2UM0ACR bit is reset to "0".

I2UM0ACR	Description
0	Receives acknowledgment "0" (initial value)
1	Receives acknowledgment "1"

- **I2UM0ER** (bit 2)
  The I2UM0ER bit is a flag to indicate a transmit error when the Master mode is selected. When the value of the bit transmitted and the value of the SDA pin do not coincide, "1" is set to this bit. When clock synchronization is used (I2UM0SYN = "1"), the SDA pin output is disabled until the subsequent byte data communication terminates, if "1" is set to the I2UM0ER bit. When clock synchronization is not used (I2UM0SYN = "0"), the SDA pin output continues until the subsequent byte data communication terminates, even if "1" is set to the I2UM0ER bit. When "1" is written to the I2UM0ER bit or "0" is written to the I2UM0EN bit, the I2UM0ER bit is reset to "0".

I2UM0ER	Description
0	No transmit error (initial value)
1	Transmit error

[Note]
Do not update each bit of the I2UM0STA bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.

12.2.9 I²C Bus 0 Receive Register (Slave) (I2US0RD)

Address: 0xF6CE
Access: R
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2US0RD							
Bit symbol	-	-	-	-	-	-	-	-	I2US0R7	I2US0R6	I2US0R5	I2US0R4	I2US0R3	I2US0R2	I2US0R1	I2US0R0
Access type	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

I2US0RD is a read-only special function register (SFR) used to store the received data when the Slave mode is selected. I2US0RD is updated after completion of each reception when the Slave mode is selected.

Description of bits

- **I2US0R7 to I2US0R0** (bits 7-0)
  The I2US0R7 to I2US0R0 bits are used to store the received data when the Slave mode is selected. The signal input to the SDA pin is received at reception of a slave address and at data transmission/reception in sync with the rising edge of the signal on the SCL pin. Since data that has been output to the SDA and SCL pins is received not only at data reception but also at data transmission, it can be checked whether transmit data has certainly been transmitted.

12.2.10 I²C Bus 0 Slave Address Register (Slave) (I2US0SA)

Address: 0xF6D0
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2US0SA							
Bit symbol	-	-	-	-	-	-	-	-	I2US0A6	I2US0A5	I2US0A4	I2US0A3	I2US0A2	I2US0A1	I2US0A0	-
Access type	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

I2US0SA is a special function register (SFR) used to set the slave address when the Slave mode is selected.

Description of bits

- **I2US0A6 to I2US0A0** (bits 7-1)
  The I2US0A6 to I2US0A0 bits are used to set the slave address when the Slave mode is selected.

12.2.11 I²C Bus 0 Transmit Data Register (Slave) (I2US0TD)

Address: 0xF6D2
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2US0TD							
Bit symbol	-	-	-	-	-	-	-	-	I2US0T7	I2US0T6	I2US0T5	I2US0T4	I2US0T3	I2US0T2	I2US0T1	I2US0T0
Access type	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

I2US0TD is a special function register (SFR) used to set the transmit data when the Slave mode is selected.

Description of bits

- **I2US0T7 to I2US0T0** (bits 7-0)
  The I2US0T7 to I2US0T0 bits are used to set the transmit data when the Slave mode is selected.

12.2.12 I²C Bus 0 Control Register (Slave) (I2US0CON)

Address: 0xF6D4
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	I2US0CON															
Byte symbol	–								I2US0CON0							
Bit symbol	-	-	-	-	-	-	-	-	I2US0ACT	-	I2US0WT	-	-	-	-	-
Access type	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

I2US0CON is a special function register (SFR) used to control transmit and receive operations when the Slave mode is selected.

Description of bits

- **I2US0WT** (bit 5)
  The I2US0WT bit is used to release the communication wait state ("L" level output on the SCL pin) when the Slave mode is selected. Writing "1" to this bit during the communication wait state releases it ("L" level output of the SCL pin is released). The I2US0WT bit is a write-only bit and "0" is always read.

I2US0WT	Description
0	Does not release the communication wait state (initial value)
1	Releases the communication wait state

- **I2US0ACT** (bit 7)
  The I2US0ACT bit is used to set the acknowledgment signal to be output at completion of reception when the Slave mode is selected.

I2US0ACT	Description
0	Acknowledgment data "0" (initial value)
1	Acknowledgment data "1"

[Note]
- Switch the system clock to the high-speed clock if releasing the communication wait status.

12.2.13 I²C Bus 0 Mode Register (Slave) (I2US0MD)

Address: 0xF6D6
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2US0MD							
Bit symbol	-	-	-	-	-	-	-	-	-	I2US0SIE	I2US0PIE	-	-	-	-	I2US0EN
Access type	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2US0MD is a special function register (SFR) used to set the operation mode when the Slave mode is selected.

Description of bits

- **I2US0EN** (bit 0)
  I2US0EN is a bit used to enable the slave operation of the I²C but unit. When "1" is written to the I2US0EN bit, the operation of the I²C bus 0 is enabled. When "0" is written to the I2US0EN bit, all the bits of the I²C bus status register (I2US0STA) are initialized to "0", and the operation of the I²C bus 0 is stopped.

I2US0EN	Description
0	Stops I²C slave operation (initial value)
1	Enables I²C slave operation

- **I2US0PIE** (bit 5)
  The I2US0PIE bit is used to select whether to disable or enable the stop condition interrupt when the Slave mode is selected.

I2US0PIE	Description
0	Disables the stop condition interrupt (initial value)
1	Enables the stop condition interrupt

- **I2US0SIE** (bit 6)
  The I2US0SIE bit is used to select whether to disable or enable the start condition interrupt when the Slave mode is selected.

I2US0SIE	Description
0	Disables the start condition interrupt (initial value)
1	Enables the start condition interrupt

[Note]
- Set I2US0EN bit to “0” to stop the operation before entering STOP/STOP-D mode.

12.2.14 I²C Bus 0 Status Register (Slave) (I2US0STA)

Address: 0xF6D8
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2US0STA							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	I2US0TR	I2US0SAA	I2US0ER	I2US0ACR	I2US0BB
Access type	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

I2US0STA is a special function register (SFR) to indicate the state of the I²C bus unit when the Slave mode is selected.

Description of bits

- **I2US0BB** (bit 0)
  The I2US0BB bit is used to indicate the state of use of the I²C bus when the Slave mode is selected. When the start condition is generated on the I²C bus, this bit is set to "1" and when the stop condition is generated, the bit is reset to "0".
  When "1" is written to the I2US0BB bit or "0" is written to the I2US0EN bit, the I2US0BB bit is reset to "0".

I2US0BB	Description
0	I²C bus-free state (initial value)
1	I²C bus-busy state

- **I2US0ACR** (bit 1)
  The I2US0ACR bit is used to store the acknowledgment signal received when the Slave mode is selected. Acknowledgment signals are received each time the slave address is received and data transmission or reception is completed.
  When "1" is written to the I2US0ACR bit or "0" is written to the I2US0EN bit, the I2US0ACR bit is reset to "0".

I2US0ACR	Description
0	Receives acknowledgment "0" (initial value)
1	Receives acknowledgment "1"

- **I2US0ER** (bit 2)
  The I2US0ER bit is a flag to indicate a transmit error when the Slave mode is selected. When the value of the bit transmitted and the value of the SDA pin do not coincide, this bit is set to "1". When I2US0ER is set to "1", the SDA pin output is disabled until the subsequent byte data communication terminates.
  When "1" is written to the I2US0ER bit or "0" is written to the I2US0EN bit, the I2US0ER bit is reset to "0".

I2US0ER	Description
0	No transmit error (initial value)
1	Transmit error

- **I2US0SAA** (bit 3)
  The I2US0SAA bit indicates that this device is specified as a slave address when the Slave mode is selected. It is set to "1" when the content of the slave address output by the master device coincides with the one of the I2US0SA register and reset to "0" when a stop condition is received.
  When "1" is written to the I2US0SAA bit or "0" is written to the I2US0EN bit, the I2US0SAA bit is reset to "0".

I2US0SAA	Description
0	Does not coincide with the slave address (initial value)
1	Coincides with the slave address

- **I2US0TR** (bit 4)
  The I2US0TR bit indicates the transmission/reception state when the Slave mode is selected. This bit is set to "1" when "1" is detected in the transfer direction specification bit. It is reset to "0" when a stop condition is detected or when "0" is detected in the transfer direction specification bit.
  The I2US0TR bit is reset to "0" by writing “0” to I2US0EN bit.

I2US0TR	Description
0	Receiving state (initial value)
1	Transmitting state

[Note]
Do not update each bit of the I2UM0STA register by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.

12.3 Description of Operation

12.3.1 Master Operation

12.3.1.1 Communication Operation Mode

Communication is started when communication mode is selected by using the I^2C bus n mode register (I2UM0MOD) after the Master mode is selected by using the I^2C bus unit n mode register (I2UnMSS), the I^2C function is enabled by using the I2UMnEN bit, a slave address and a data communication direction are set in the I^2C bus n slave address register (I2UMnSA), and "1" is written to the I2UMnST bit of the I^2C bus n control register (I2UMnCON).

12.3.1.2 Start Condition

When "1" is written to the I2UMnST bit of the I^2C bus n control register (I2UMnCON) while communication is stopped (the I2UMnST bit is "0"), communication is started and the start condition waveform is output to the SDA and SCL pins. After execution of the start condition, the LSI shifts to the slave address transmit mode.

12.3.1.3 Restart Condition

When "1" is written to the I2UMnRS and I2UMnST bits of the I^2C bus n control register (I2UMnCON) during communication (the I2UMnST bit is "1"), the restart condition waveform is output to the SDA and SCL pins.
After execution of the restart condition, the LSI shifts to the slave address transmit mode.

12.3.1.4 Slave Address Transmit Mode

In the slave address transmit mode, the values (slave address and data communication direction) of the I^2C bus n slave address register (I2UMnSA) are transmitted in MSB first, and finally, the acknowledgment signal is received in the I2UMnACR bit of the I^2C bus n status register (I2UMnSTA).
At completion of acknowledgment reception, the LSI shifts to the I^2C bus n control register (I2UMnCON) setting wait state (control register setting wait state).
The values of I2UMnSA output from the SDA pin are stored in I2UMnRD.

12.3.1.5 Data Transmit Mode

In the data transmit mode, the value of I2UMnTD is transmitted in MSB first, and finally, the acknowledgment signal is received in the I2UMnACR bit of the I^2C bus n status register (I2UMnSTA).
At completion of acknowledgment reception, the LSI shifts to the I^2C bus control register (I2UMnCON) setting wait state (control register setting wait state).
The values of I2UMnTD output from the SDA pin are stored in I2UMnRD.

12.3.1.6 Data Receive Mode

In the data receive mode, the value input in the SDA pin is received synchronously with the rising edge of the serial clock input to the SCL pin, and finally, the value of the I2UMnACT bit of the I^2C bus n control register (I2UMnCON) is output.
At completion of acknowledgment transmission, the LSI shifts to the I^2C bus n control register (I2UMnCON) setting wait state (control register setting wait state).
The data received is stored in I2UMnRD after the acknowledgment signal is output. The acknowledgment signal output is received in the I2UMnACR bit of the I^2C bus n status register (I2UMnSTA).

12.3.1.7 Control Register Setting Wait State

When the LSI shifts to the control register setting wait state, an I^2C bus unit interrupt (I2UMnINT) is output.
In the control register setting wait state, the transmit error flag (I2UMnER) of the I^2C bus n status register (I2UMnSTA) and acknowledgment receive data (I2UMnACR) are confirmed and at data reception, the contents of I2UMnRD are read in the CPU and the next operation mode is selected.
When "1" is written to the I2UMnST bit in the control register setting wait state, the LSI shifts to the data transmit or receive mode. When "1" is written to the I2UMnSP bit, the LSI shifts to the stop condition. When "1" is written to the I2UMnRS bit, the operation shifts to the restart condition.

12.3.1.8 Stop Condition

In the stop condition, the stop condition waveform is output to the SDA and SCL pins. After the stop condition waveform is output, an I^2C bus unit interrupt (I2CUnINT) is generated.

12.3.2 Master Mode Communication Operation Timing

Figures 12-2 to 12-4 show the operation timing and control method for each communication mode during the master operation.

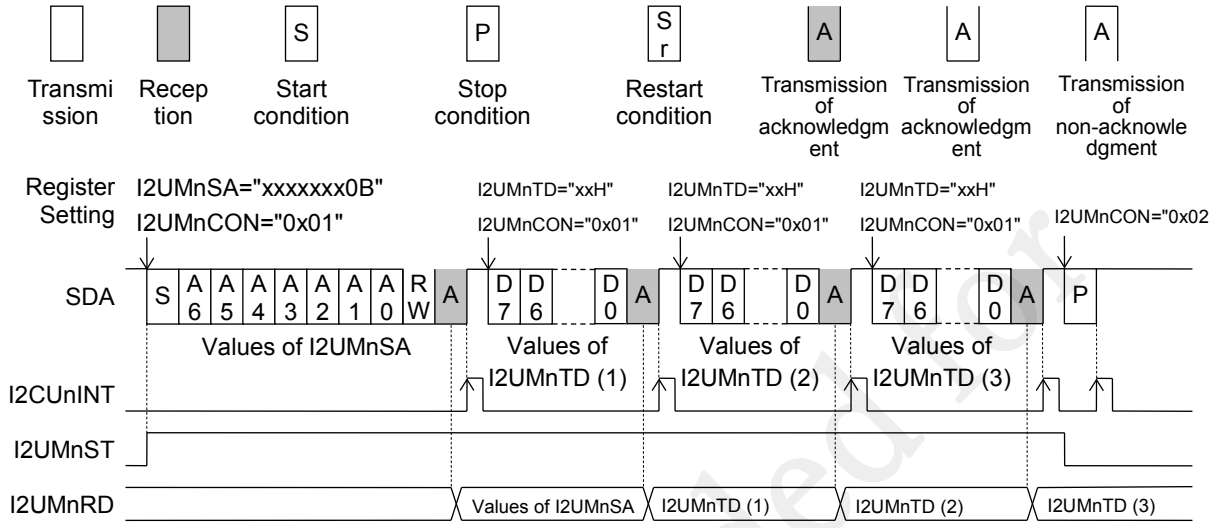


Figure 12-2 Operation Timing in Data Transmit Mode When Master Mode is Selected

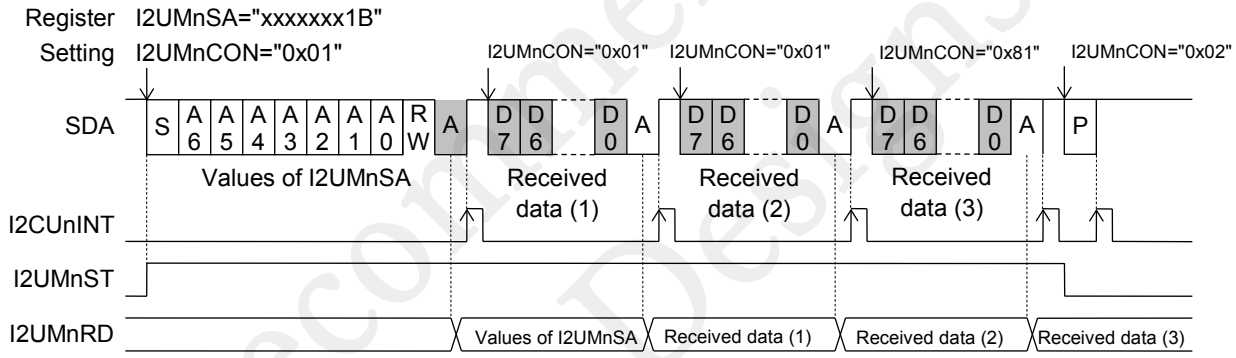


Figure 12-3 Operation Timing in Data Receive Mode When Master Mode is Selected

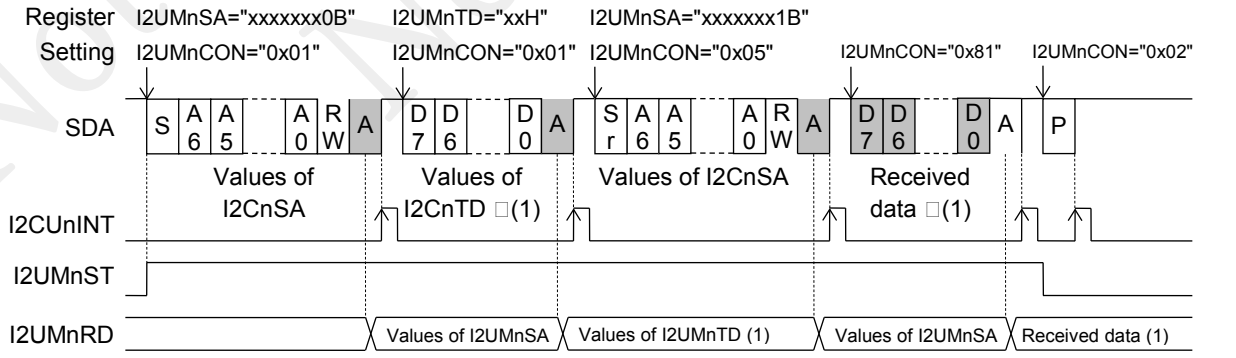


Figure 12-4 Operation Timing at Data Transmit/Receive Mode Switching When Master Mode is Selected

Figure 12-5 shows the operation timing and control method when an acknowledgment error occurs.

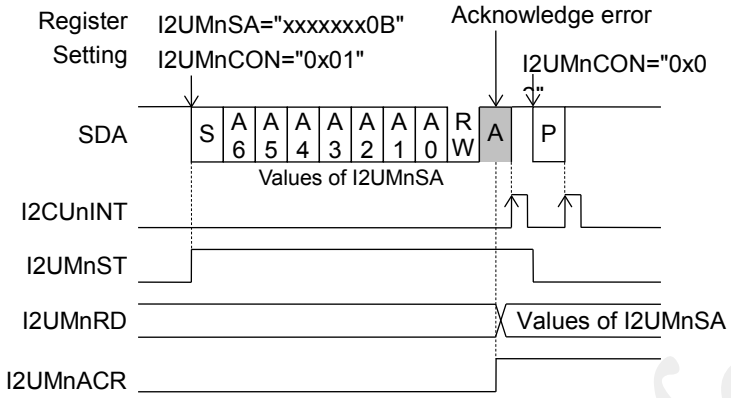


Figure 12-5 Operation Suspend Timing at Occurrence of Acknowledgment Error When Master Mode Is Selected

Figure 12-6 shows the operation timing and control method when transmission fails.

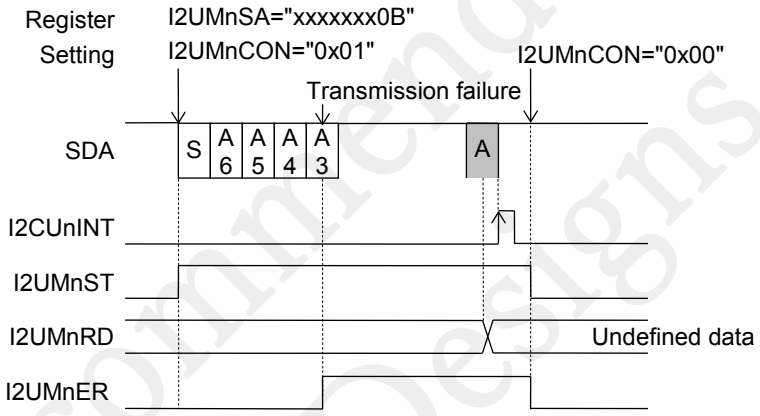


Figure 12-6 Operation Timing When Transmission Fails When Master Mode is Selected

12.3.3 Slave Operation

12.3.3.1 Communication Operation Mode

When a slave address is set in the I^2C bus n slave address register (I2USnSA) after the Slave mode is selected by using the I^2C bus unit n mode register (I2UnMSS), the start condition interrupt and stop condition interrupt are enabled by the I^2C bus n mode register (I2USnMD), and "1" is written to the I2USnEN bit, then the receive operation is enabled.

12.3.3.2 Start Condition

When the start condition waveform is input to the SDA and SCL pins, the I2USnBB bit of the I^2C bus n status register (I2USnSTA) is set to "1", and the receive operation is started. After the start condition is finished, the LSI shifts to the slave address receive mode.
When the start condition interrupt is enabled by the I2USnSIE bit of the I^2C bus n mode register (I2USnMD), an I^2C bus n interface interrupt (I2CUnINT) is output.

12.3.3.3 Slave Address Receive Mode

In the slave address receive mode, the values input in the SDA pin (slave address, transfer direction specification bit) are received synchronously with the rising edge of the serial clock input to the SCL pin.
When the value of the received slave address coincides with the one set in the I^2C bus n slave address register (I2USnSA), the I2USnSAA bit of the^2C bus n status register (I2USnSTA) is set to "1". Then, the value of the received data communication direction is stored in the I2USnTR bit of I2USnSTA, and finally the acknowledgment data ("L" level) is output. After a falling edge of the SCL pin is detected during the acknowledgment data transmission, the system enters the communication wait state and an I^2C bus n interface interrupt (I2CUnINT) is output while the acknowledgment data is transmitted.
When the value of the received slave address does not coincide with the one set in the I^2C bus n slave address register (I2USnSA), the I2USnSAA bit remains "n". Storing data into the I2USnTR bit is not performed, nor output of the acknowledgment data, and thus the system does not enter the communication wait state. Also, an I^2C bus n interface interrupt (I2CUnINT) is not output.

12.3.3.4 Communication Wait State

In the communication wait state, the SCL pin is fixed to "L" level to put the communication in the wait state.
In the data receive mode, when the next data reception is ready, "1" is written into the I2USnWT bit of the I^2C bus n control register (I2USnCON) to release the communication wait state.
In the data transmit mode, after the next data to be transmitted is set to the I^2C bus n transmit data register (I2USnTD), "1" is written into the I2USnWT bit of I2USnCON to release the communication wait state.

12.3.3.5 Data Transmit Mode

In the data transmit mode, the value of I2USnTD is transmitted in MSB first, and finally, the acknowledgment signal is received in the I2USnACR bit of the I^2C bus n status register (I2USnSTA).
After the falling edge of the transfer clock input in the SCL pin is detected while receiving the acknowledgment data, the system enters the communication wait state, and an I^2C bus 1 interface interrupt (I2CUnINT) is output while the acknowledgment data is received.
The values of I2USnTD output from the SDA pin are stored in I2USnRD.

12.3.3.6 Data Receive Mode

In the data receive mode, the value input in the SDA pin is received synchronously with the rising edge of the serial clock input to the SCL pin, and finally, the value (acknowledgment signal) of the I2USnACT bit of the I^2C bus n control register (I2USnCON) is output.
After the falling edge of the transfer clock input in the SCL pin is detected while receiving the acknowledgment data, the system enters the communication wait state, and an I^2C bus n interface interrupt (I2CUnINT) is output while the acknowledgment data is transmitted.
The received data is stored in I2USnRD. The acknowledgment signal output is received in the I2USnACR bit of the I^2C bus n status register (I2USnSTA).

12.3.3.7 Stop Condition

When the stop condition waveform is input to the SDA and SCL pins, the I2USnBB bit of the I^2C bus n status register (I2USnSTA) is reset to "n", and the receive operation is stopped.
When the stop condition interrupt is enabled by the I2USnPIE bit of the I2C bus n mode register (I2USnMD), an I2C bus n interface interrupt (I2CUnINT) is output.

12.3.4 Communication Operation Timing

Figures 12-7 to 12-9 show the operation timing and control method for each communication mode.

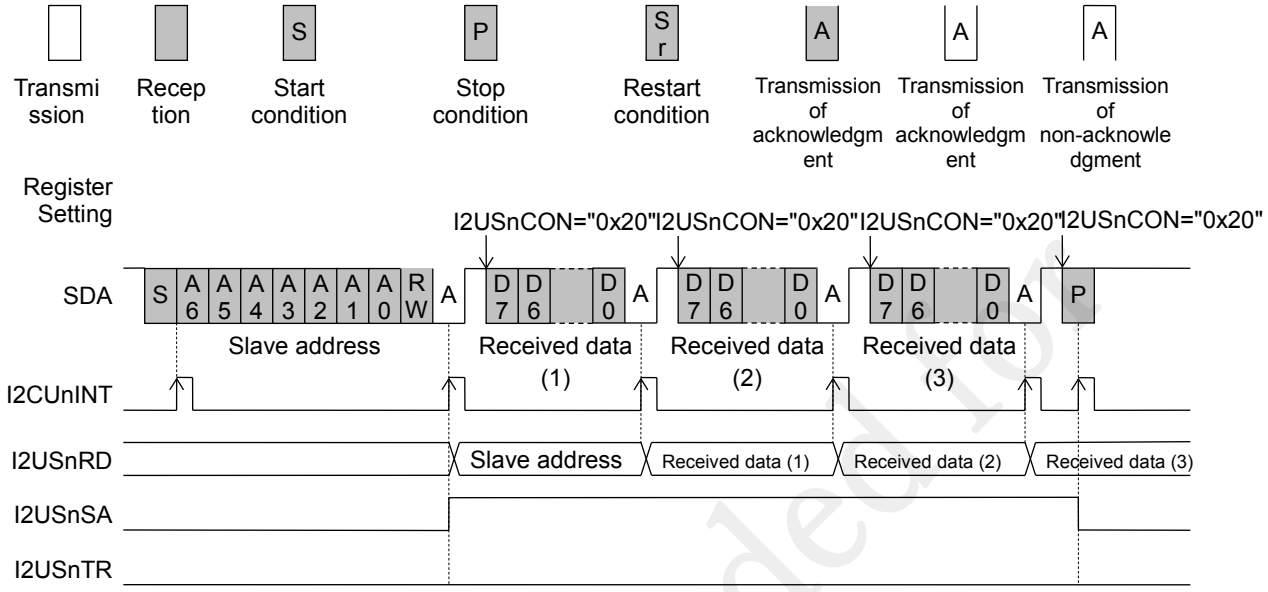


Figure 12-7 Operation Timing in Data Receive Mode When Slave Mode is Selected

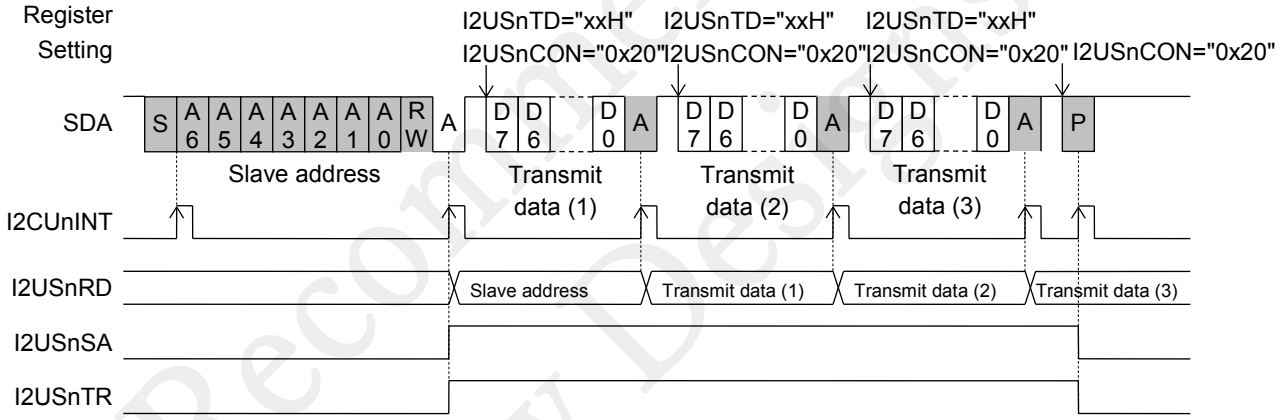


Figure 12-8 Operation Timing in Data Transmit Mode When Slave Mode is Selected

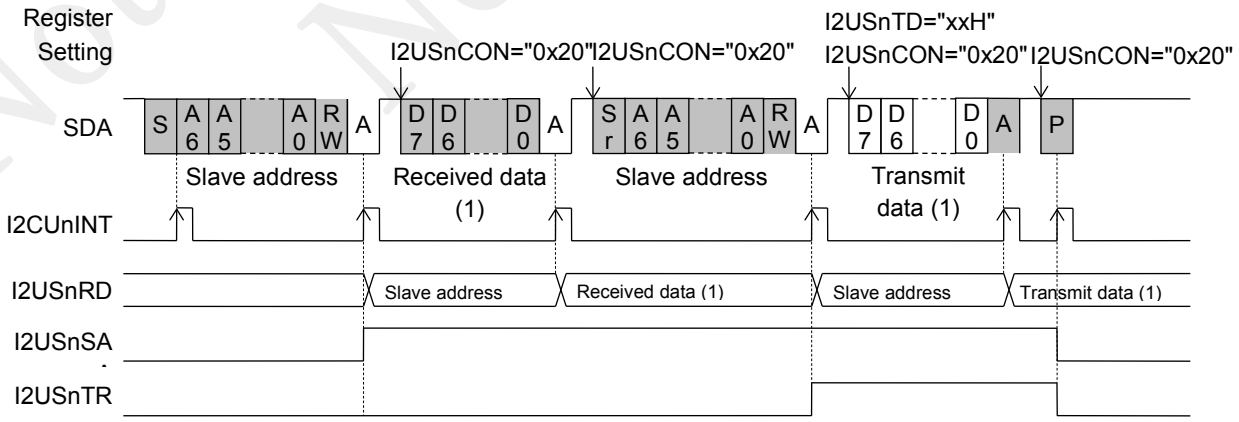


Figure 12-9 Operation Timing at Data Transmit/Receive Mode Switching When Slave Mode is Selected

When the values of the transmitted bit and the SDA pin do not coincide, the I2USnER bit of the I^{2}C bus n status register (I2USnSTA) is set to "1" and the SDA pin output is disabled until termination of the subsequent byte data communication.
Figure 12-10 shows the operation timing and control method when transmission fails.

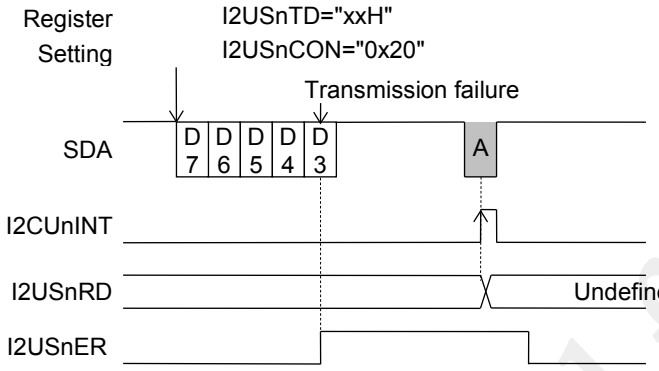


Figure 12-10 Operation Timing When Transmission Fails When Slave Mode is Selected

When the value of the received slave address does not coincide with the one set in the I^{2}C bus n slave address register (I2USnSA), the I2USnSAA bit remains "n". Storing data into the I2USnTR bit is not performed, nor output of the acknowledgment data, and thus the system does not enter the communication wait state. Also, an I^{2}C bus n interface interrupt (I2CUnINT) is not output.
Figure 12-11 shows the operation timing when the slave address does not match.

Figure 12-11 Operation Timing When Slave address does not match in the Slave mode

12.3.5 Operation Waveforms

Figure 12-11 shows the operation waveforms of the SDA and SCL signals and the I2UMnBB flag. Tables 12-1 and 12-2 show the relationship between communication speeds and HSCLK clock counts.

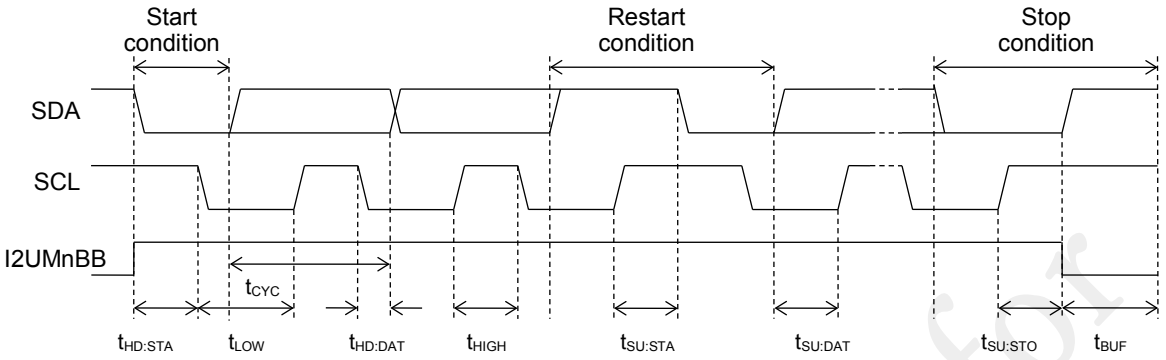


Figure 12-11 Operation Waveforms of SDA and SCL Signals and I2UMnBB Flag

Table 12-1 Relationship between Communication Speeds and HSCLK Clock Counts (HSCLK = 24 MHz)

Communication speed (I2UMnMD)	Speed reduction (I2UMnDW1, 0)	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Standard mode 100 kbps	No reduction	240φ	108φ	132φ	24φ	108φ	132φ	108φ	108φ	132φ
	10% reduction	264φ	120φ	144φ	24φ	120φ	144φ	120φ	120φ	144φ
	20% reduction	288φ	132φ	156φ	24φ	132φ	156φ	132φ	132φ	156φ
	30% reduction	312φ	144φ	168φ	24φ	144φ	168φ	144φ	144φ	168φ
Fast mode 400kbps	No reduction	60φ	24φ	36φ	12φ	24φ	36φ	24φ	24φ	36φ
	10% reduction	66φ	27φ	39φ	12φ	27φ	39φ	27φ	27φ	39φ
	20% reduction	72φ	30φ	42φ	12φ	30φ	42φ	30φ	30φ	42φ
	30% reduction	78φ	33φ	45φ	12φ	33φ	45φ	33φ	33φ	45φ
1Mbps mode 1Mbps	No reduction	24φ	10φ	14φ	4φ	10φ	14φ	10φ	10φ	14φ
	10% reduction	26φ	11φ	15φ	4φ	11φ	15φ	11φ	11φ	15φ
	20% reduction	29φ	13φ	16φ	4φ	13φ	16φ	12φ	13φ	16φ
	30% reduction	31φ	14φ	17φ	4φ	14φ	17φ	13φ	14φ	17φ

The above clock counts are values when HSCLK is selected for the operating frequency (I2mnCD[2:0]="000"). When 1/2 or 1/4HSCLK is selected, the counts increase in proportion to the dividing ratio.
Specify the I^2C operation clock as follows.

When the HSCLK is 24MHz
- Standard mode : I^2C operation clock HSCLK~1/4HSCLK
- Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
- 1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK

φ: Clock cycle of 1/m HSCLK (m is selected by the I2mnCD[2:0] bits of the I2UMnMOD register)

Table 12-2 Relationship between Communication Speeds and HSCLK Clock Counts (HSCLK = 16 or 32 MHz)

Communication speed (I2UMnMD)	Speed reduction (I2UMnDW1, 0)	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Standard mode 100 kbps	No reduction	160φ	72φ	88φ	16φ	72φ	88φ	72φ	72φ	88φ
	10% reduction	176φ	80φ	96φ	16φ	80φ	96φ	80φ	80φ	96φ
	20% reduction	192φ	88φ	104φ	16φ	88φ	104φ	88φ	88φ	104φ
	30% reduction	208φ	96φ	112φ	16φ	96φ	112φ	96φ	96φ	112φ
Fast mode 400kbps	No reduction	40φ	14φ	26φ	12φ	14φ	26φ	14φ	14φ	26φ
	10% reduction	44φ	16φ	28φ	12φ	16φ	28φ	16φ	16φ	28φ
	20% reduction	48φ	18φ	30φ	12φ	18φ	30φ	18φ	18φ	30φ
	30% reduction	52φ	20φ	32φ	12φ	20φ	32φ	20φ	20φ	32φ
1Mbps mode 1Mbps	No reduction	16φ	6φ	10φ	4φ	6φ	10φ	6φ	6φ	10φ
	10% reduction	18φ	7φ	11φ	4φ	7φ	11φ	7φ	7φ	11φ
	20% reduction	19φ	8φ	11φ	4φ	8φ	11φ	7φ	8φ	11φ
	30% reduction	21φ	9φ	12φ	4φ	9φ	12φ	8φ	9φ	12φ

The above clock counts are values when HSCLK is selected for the operating frequency (I2mnCD[2:0]="000"). When 1/2 or 1/4HSCLK is selected, the counts increase in proportion to the dividing ratio.
Specify the I^2C operation clock as follows.

When the HSCLK is 16MHz
- Standard mode : I^2C operation clock HSCLK, 1/2HSCLK
- Fast mode : I^2C operation clock HSCLK
- 1Mbps mode : I^2C operation clock HSCLK

When the HSCLK is 32MHz
- Standard mode : I^2C operation clock HSCLK~1/4HSCLK
- Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
- 1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK

φ: Clock cycle of 1/m HSCLK (m is selected by the I2mnCD[2:0] bits of the I2UMnMOD register)

Table 12-3 Relationship between Communication Speeds and LSCLK Clock Counts

I2UMnMD1	I2UMnMD0	Communication speed	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
0	0	2.048kbps	16φ	8φ	8φ	1φ	8φ	8φ	7φ	8φ	8φ
0	1	4.096kbps	8φ	4φ	4φ	1φ	4φ	4φ	3φ	4φ	4φ
1	*	8.192kbps	4φ	2φ	2φ	1φ	2φ	2φ	1φ	2φ	2φ

The above clock counts are values when LSCLK is selected for the operating frequency (I2mnCD[2:0]="000"). When 1/2 to 1/8LSCLK is selected, the counts increase in proportion to the dividing ratio.
φ: Clock cycle of 1/m LSCLK (m is selected by the I2mnCD[2:0] bits of the I2UMnMOD register)
When LSCLK is used as the clock, the setting value of I2UMnDW1 and 0 is ignored.

[Note]
When the slave device uses the clock synchronization function (handshake function) which holds the SCL signal at a "L" level, the time t_{CYC} and time t_{LOW} are extended.

12.3.6 Pin Settings

SCL and SDA can be selected from multiple GPIOs.
Be sure to select one of the following combinations of GPIOs for SCL and SDA.

Function	I/O pin	Combination 1	Combination 2	Combination 3
I^2C Unit	I2CUn_SDA	P03	P15	P26
	I2CUn_SCL	P04	P16	P27

GPIOs used for SDA and SCL must be set. In addtion to the mode setting, select to enable intput, enable output, Nch open drain output and not pull-up.

Byte symbol	PnMODm							
Bit symbol	Pn0MD3	Pn0MD2	Pn0MD1	Pn0MD0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
Setting value	0	0	1	1	1	0	1	1

n: Port number (n=0~3)
m: Bit number (n=0~7)

[Note]
Use external pull-up resistors for SDA pin and SCL pin refering to the I^2C bus specification. The internal pull-up resistors can not satisfy the I^2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.

Chapter 13 I^2C Master

13. I²C Master

13.1 General Description

ML62Q1000 series has one channel of the I²C master. The I²C master has master-only function on which the slave function and the low-speed clock (LSCLK) operation are removed from the I²C bus unit described in Chapter 12.

13.1.1 Features

- Master function
- Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1 Mbps) are supported for the communication speed
- Clock synchronization (handshake) can be supported
- 7-bit address format (10bit address format is supported)
- Self-test function (safety function)

13.1.2 Configuration

Figure 13-1 shows the configuration diagram of the I²C master circuit.

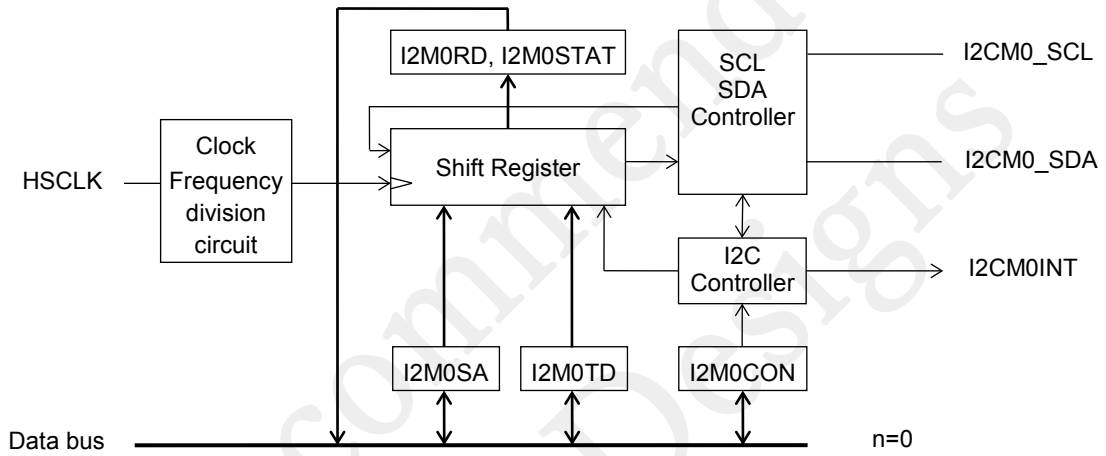


Figure 13-1 Configuration Diagram of I²C Master

I2M0RD : I²C master 0 receive register
I2M0SA : I²C master 0 slave address register
I2M0TD : I²C master 0 transmit data register
I2M0CON : I²C master 0 control register
I2M0MOD : I²C master 0 mode register
I2M0STAT : I²C master 0 status register

[Note]
- This I²C master unit does not support the multiple master devices on the I²C bus.

13.1.3 List of Pins

The I/O pins of the I^2C master are assigned to the secondary to octic functions of the GPIO.
For details about pin assignment and how to set the secondary to octic functions of the GPIO, see Chapter 17 "GPIO".

Pin name	I/O	Function
I2CM0_SDA	I/O	I^2C master 0 data I/O pin
I2CM0_SCL	I/O	I^2C master 0 clock I/O pin

13.2 Description of Registers

13.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF6E0	Reserved register	-	-	R	8	0x00
0xF6E1	Reserved register	-	-	R	8	0x00
0xF6E2	I^2C master 0 receive register	I2M0RD	-	R	8	0x00
0xF6E3	Reserved register	-	-	R	8	0x00
0xF6E4	I^2C master 0 slave address register	I2M0SA	-	R/W	8	0x00
0xF6E5	Reserved register	-	-	R	8	0x00
0xF6E6	I^2C master 0 transmit data register	I2M0TD	-	R/W	8	0x00
0xF6E7	Reserved register	-	-	R	8	0x00
0xF6E8	I^2C master 0 control register	I2M0CON	-	R/W	8	0x00
0xF6E9	Reserved register	-	-	R	8	0x00
0xF6EA	I^2C master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF6EB		I2M0MODH		R/W	8	0x02
0xF6EC	I^2C master 0 status register	I2M0STAT	-	R/W	8	0x00
0xF6ED	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

13.2.2 I²C Master 0 Receive Register (I2M0RD)

Address: 0xF6E2
Access: R
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2M0RD							
Bit symbol	-	-	-	-	-	-	-	-	I2M0R7	I2M0R6	I2M0R5	I2M0R4	I2M0R3	I2M0R2	I2M0R1	I2M0R0
Access type	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

I2MnRD is a read-only special function register (SFR) used to store the received data.
I2MnRD is updated after completion of each reception.

Description of bits

- **I2M0R7 to I2M0R0** (bits 7-0)
  The I2M0R7 to I2M0R0 bits are used to store the received data. The signal input to the SDA pin is received at transmission of a slave address and at data transmission/reception in sync with the rising edge of the signal on the SCL pin. Since data that has been output to the SDA and SCL pins is received not only at data reception but also at slave address data transmission and data transmission, it can be checked whether transmit data has certainly been transmitted.

13.2.3 I²C Master 0 Slave Address Register (I2M0SA)

Address: 0xF6E4
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2M0SA							
Bit symbol	-	-	-	-	-	-	-	-	I2M0A6	I2M0A5	I2M0A4	I2M0A3	I2M0A2	I2M0A1	I2M0A0	I2M0RW
Access type	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

I2M0SA is a special function register (SFR) to set the address and transmit/receive mode of the slave device.

Description of bits

- **I2M0RW** (bit 0)
  The I2MnRW bit is used to select the data receive mode.

I2M0RW	Description
0	Data transmit mode (initial value)
1	Data Receive Mode

- **I2M0A6 to I2M0A0** (bits 7-1)
  The I2M0A6 to I2M0A0 bits are used to set the slave address.

13.2.4 I²C Master 0 Transmit Data Register (I2M0TD)

Address: 0xF6E6
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2M0TD							
Bit symbol	-	-	-	-	-	-	-	-	I2M0T7	I2M0T6	I2M0T5	I2M0T4	I2M0T3	I2M0T2	I2M0T1	I2M0T0
Access type	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

I2M0TD is a special function register (SFR) used to set the transmit data.

Description of bits

- **I2M0T7 to I2M0T0** (bits 7-0)
  The I2M0T7 to I2M0T0 bits are used to set the transmit data.

13.2.5 I²C Master 0 Control Register (I2M0CON)

Address: 0xF6E8
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2M0CON							
Bit symbol	-	-	-	-	-	-	-	-	I2M0ACT	-	-	-	-	I2M0RS	I2M0SP	I2M0ST
Access type	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

I2M0CON is a special function register (SFR) used to control transmit and receive operations.

Description of bits

- **I2M0ST** (bit 0)
  The I2M0ST bit is used to control the communication operation of the I²C master. When "1" is written to the I2M0ST bit, communication is started.
  When "1" is overwritten to the I2M0ST bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, data transmission/reception restarts. When "0" is written to the I2M0ST bit, communication stops forcibly.
  "1" can be written to the I2M0ST bit only when the I²C master is in an operation enable state (I2M0EN = "1").
  When "1" is written to the I2M0SP bit, the I2M0ST bit is reset to "0".

I2M0ST	Description
0	Stops communication (initial value)
1	Starts communication

- **I2M0SP** (bit 1)
  The I2M0SP bit is a write-only bit used to request a stop condition.
  When "1" is written to the I2M0SP bit, the bus shifts to the stop condition and communication stops.
  When the I2M0SP bit is read, "0" is always read.

I2M0SP	Description
0	No stop condition request (initial value)
1	Stop condition request

- **I2M0RS** (bit 2)
  The I2MnRS bit is a write-only bit used to request a restart. When "1" is written to this bit during data communication, the LSI shifts to the restart condition and communication restarts from the slave address. "1" can be written to I2M0RS only while communication is active (I2M0ST = "1").
  When the I2M0RS bit is read, "0" is always read.

I2M0RS	Description
0	No restart request (initial value)
1	Restart request

- **I2M0ACT** (bit 7)
  The I2MnACT bit is used to set the acknowledgment data to be output at completion of reception.

I2M0ACT	Description
0	Acknowledgment data "0" (initial value)
1	Acknowledgment data "1"

[Note]
- Do not update the I2MnACT bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.
- When the I2MnST bit is “1”, write the I2MnCON register in the control register setting wait state.

13.2.6 I²C Master 0 Mode Register (I2M0MOD)

Address: 0xF6EA
Access: R/W
Access size: 8/16 bits
Initial value: 0x0200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	I2M0MOD															
Byte symbol	I2M0MODH								I2M0MODL							
Bit symbol	-	-	-	-	-	-	I2M0CD1	I2M0CD0	-	-	I2M0SYN	I2M0DW1	I2M0DW0	I2M0MD1	I2M0MD0	I2M0EN
Access type	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

I2M0MOD is a special function register (SFR) used to set the operation mode.

Description of bits

- **I2M0EN** (bit 0)
  The I2M0EN bit is used to enable the master operation of the I²C master n. When "1" is written to the I2M0EN bit, the I2M0ST bit can be set and the I2M0BB flag starts operation. When "0" is written to I2M0EN, the I²C master stops operation and the I2UM0RD, I2UM0SA, I2UM0TD, and I2UM0CON registers are initialized.

I2M0EN	Description
0	Stops I²C Master operation (initial value)
1	Enables I²C Master operation

- **I2M0MD1 to I2M0MD0** (bits 2-1)
  The I2M0MD1 to I2M0MD0 bits are used to set the communication speed of the I²C master. The standard mode, fast mode, or high speed mode can be selected.

I2M0MD1	I2M0MD0	Description
0	0	Standard mode (initial value)/100 kbps*
0	1	Fast mode/400 kbps*
1	0	High speed mode/1 Mbps*
1	1	High speed mode/1 Mbps*

* : I2MnCD[1:0] is "00" & I2MnSYN is "0"

- **I2M0DW1 to I2M0DW0** (bits 4-3)
  The I2M0DW1 to I2M0DW0 bits are used to set the communication speed reduction rate of the I²C master. Set this bit so that the communication speed does not exceed 100 kbps/400 kbps/1 Mbps.

I2M0DW1	I2M0DW0	Description
0	0	No communication speed reduction (initial value)
0	1	10% communication speed reduction
1	0	20% communication speed reduction
1	1	30% communication speed reduction

- **I2M0SYN** (bit 5)
  The I2UM0SYN bit is used to select whether or not to use the clock synchronization (handshake) function when the Master mode is selected. Set the bit to "1" when using the clock synchronization function. By setting the bit is set to "1", monitors the I^2C bus, threfore, the communication speed gets lower depending on the load of I^2C bus.

I2M0SYN	Description
0	Disables clock synchronization (initial value)
1	Enables clock synchronization

- **I2M0CD1 to I2M0CD0** (bits 9-8)
  The I2M0CD1 to I2M0CD0 bits are used to set the operating frequency of I^2C. Set a frequency division value of HSCLK.

I2M0CD1	I2M0CD0	Description
0	0	HSCLK
0	1	1/2HSCLK
1	0	1/4HSCLK (initial value)
1	1	Do not use

[Note]
- Specify the I^2C operation clock as follows.
  When the HSCLK is 32MHz
  Standard mode : I^2C operation clock HSCLK~1/4HSCLK
  Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
  1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK
  When the HSCLK is 24MHz
  Standard mode : I^2C operation clock HSCLK~1/4HSCLK
  Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
  1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK
  When the HSCLK is 16MHz
  Standard mode : I^2C operation clock HSCLK, 1/2HSCLK
  Fast mode : I^2C operation clock HSCLK
  1Mbps mode : I^2C operation clock HSCLK

13.2.7 I²C Master 0 Status Register (I2M0STAT)

Address: 0xF6EC
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								I2M0STA							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	I2M0ER	I2M0ACR	I2M0BB
Access type	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

I2M0STAT is a special function register (SFR) to indicate the state of the I²C master.

Description of bits

- **I2M0BB** (bit 0)
  The I2M0BB bit is used to indicate the state of use of the I²C bus. When the start condition is generated on the I²C bus, this bit is set to "1" and when the stop condition is generated, the bit is reset to "0".
  When "1" is written to the I2M0BB bit, the I2M0BB bit is reset to "0".

I2M0BB	Description
0	I²C bus-free state (initial value)
1	I²C bus-busy state

- **I2M0ACR** (bit 1)
  The I2S0ACR bit is used to store the acknowledgment signal received. Acknowledgment signals are received each time the slave address is received and data transmission or reception is completed.
  When "1" is written to the I2M0ACR bit, the I2M0ACR bit is reset to "0".

I2M0ACR	Description
0	Receives acknowledgment "0" (initial value)
1	Receives acknowledgment "1"

- **I2M0ER** (bit 2)
  The I2M0ER bit is a flag to indicate a transmit error. When the value of the bit transmitted and the value of the SDA pin do not coincide, "1" is set to this bit. When clock synchronization is used (I2M0SYN = "1"), the SDA pin output is disabled until the subsequent byte data communication terminates, if "1" is set to the I2M0ER bit. When clock synchronization is not used (I2M0SYN = "0"), the SDA pin output continues until the subsequent byte data communication terminates, even if "1" is set to the I2M0ER bit.
  When "1" is written to the I2M0ER bit or "0" is written to the I2M0EN bit, the I2M0ER bit is reset to "0".

I2M0ER	Description
0	No transmit error (initial value)
1	Transmit error

13.3 Description of Operation

13.3.1 Master Operation

13.3.1.1 Communication Operation Mode

Communication is started when communication mode is selected by using the I^2C master n mode register (I2M0MOD), the I^2C function is enabled by using the I2M0EN bit, a slave address and a data communication direction are set in the I^2C bus n slave address register (I2M0SA), and "1" is written to the I2M0ST bit of the I^2C bus n control register (I2M0CON).

13.3.1.2 Start condition

When "1" is written to the I2M0ST bit of I2M0CON while communication is stopped (the I2M0ST bit is "0"), communication is started and the start condition waveform is output to the SDA and I2CM0_SCL pins.
After execution of the start condition, the LSI shifts to the slave address transmit mode.

13.3.1.3 Restart Condition

When "1" is written to the I2M0RS and I2M0ST bits of the I^2C bus n control register (I2M0CON) during communication (the I2M0ST bit is "1"), the restart condition waveform is output to the I2CM0_SDA and I2CM0_SCL pins.
After execution of the restart condition, the LSI shifts to the slave address transmit mode.

13.3.1.4 Slave Address Transmit Mode

In the slave address transmit mode, the values (slave address and data communication direction) of the I^2C bus n slave address register (I2M0SA) are transmitted in MSB first, and finally, the acknowledgment signal is received in the I2M0ACR bit of the I^2C bus n status register (I2M0STAT).
At completion of acknowledgment reception, the LSI shifts to the I^2C bus n control register (I2M0CON) setting wait state (control register setting wait state).
The values of I2M0SA output from the SDA pin are stored in I2M0RD.

13.3.1.5 Data Transmit Mode

In the data transmit mode, the value of I2M0TD is transmitted in MSB first, and finally, the acknowledgment signal is received in the I2MnACR bit of the I^2C bus n status register (I2M0STAT).
At completion of acknowledgment reception, the LSI shifts to the I^2C bus control register (I2M0CON) setting wait state (control register setting wait state).
The values of I2M0TD output from the SDA pin are stored in I2M0RD.

13.3.1.6 Data Receive Mode

In the data receive mode, the value input in the I2CM0_SDA pin is received synchronously with the rising edge of the serial clock input to the I2CM0_SCL pin, and finally, the value of the I2M0ACT bit of the I^2C bus n control register (I2M0CON) is output.
At completion of acknowledgment transmission, the LSI shifts to the I^2C bus n control register (I2M0CON) setting wait state (control register setting wait state).
The data received is stored in I2M0RD after the acknowledgment signal is output. The acknowledgment signal output is received in the I2M0ACR bit of the I^2C bus n status register (I2M0STAT).

13.3.1.7 Control Register Setting Wait State

When the LSI shifts to the control register setting wait state, an I^2C master interrupt (I2M0INT) is output.
In the control register setting wait state, the transmit error flag (I2M0ER) of the I^2C bus n status register (I2M0STAT) and acknowledgment receive data (I2M0ACR) are confirmed and at data reception, the contents of I2M0RD are read in the CPU and the next operation mode is selected.
When "1" is written to the I2M0ST bit in the control register setting wait state, the LSI shifts to the data transmit or receive mode. When "1" is written to the I2M0SP bit, the LSI shifts to the stop condition. When "1" is written to the I2M0RS bit, the operation shifts to the restart condition.

13.3.1.8 Stop Condition

In the stop condition, the stop condition waveform is output to the I2CM0_SDA and I2CM0_SCL pins. After the stop condition waveform is output, an I^2C master interrupt (I2C0INT) is generated.

13.3.2 Communication Operation Timing

Figures 13-2 to 13-4 show the operation timing and control method for each communication mode.

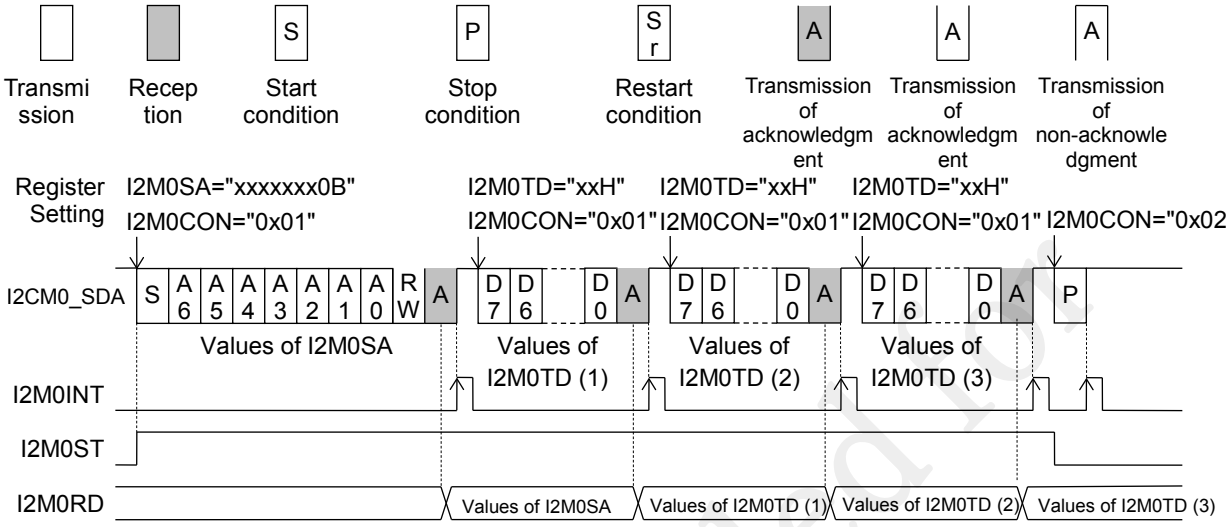


Figure 13-2 Operation Timing in Data Transmit Mode

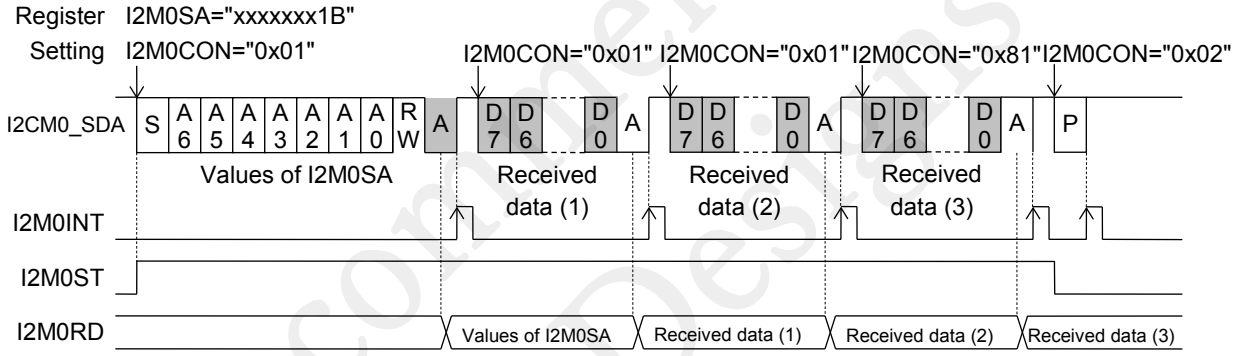


Figure 13-3 Operation Timing in Data Receive Mode

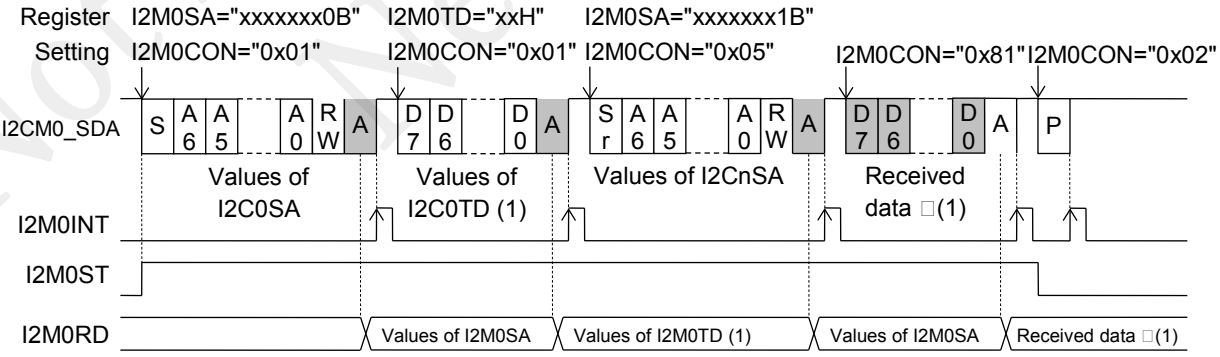


Figure 13-4 Operation Timing at Data Transmit/Receive Mode Switching

Figure 13-5 shows the operation timing and control method when an acknowledgment error occurs.

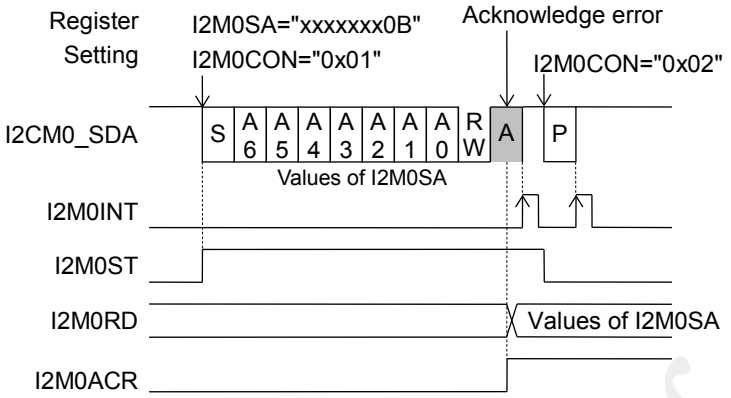


Figure 13-5 Operation Suspend Timing at Occurrence of Acknowledgment Error

Figure 13-6 shows the operation timing and control method when transmission fails.

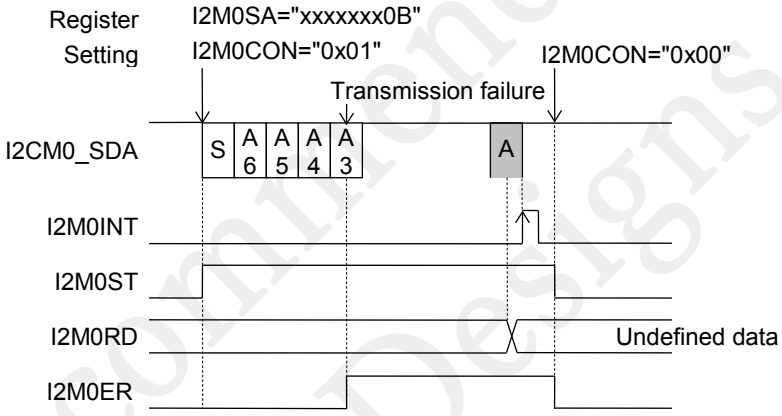


Figure 13-6 Operation Timing When Transmission Fails

13.3.3 Operation Waveforms

Figure 13-7 shows the operation waveforms of the I2CM0_SDA and I2CM0_SCL signals and the I2M0BB flag. Tables 13-1 and 13-2 show the relationship between communication speeds and HSCLK clock counts.

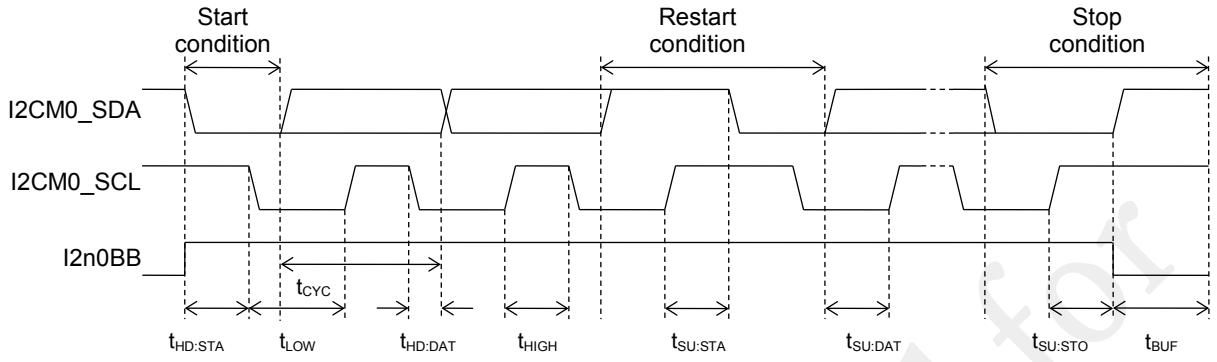


Figure 13-7 Operation Waveforms of I2CM0_SDA and I2CM0_SCL Signals and I2n0BB Flag

Table 13-1 Relationship between Communication Speeds and HSCLK Clock Counts (HSCLK = 24 MHz)

Communication speed (I2M0SP)	Speed reduction (I2M0DW1, 0)	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Standard mode 100 kbps	No reduction	240φ	108φ	132φ	24φ	108φ	132φ	108φ	108φ	132φ
	10% reduction	264φ	120φ	144φ	24φ	120φ	144φ	120φ	120φ	144φ
	20% reduction	288φ	132φ	156φ	24φ	132φ	156φ	132φ	132φ	156φ
	30% reduction	312φ	144φ	168φ	24φ	144φ	168φ	144φ	144φ	168φ
Fast mode 400kbps	No reduction	60φ	24φ	36φ	12φ	24φ	36φ	24φ	24φ	36φ
	10% reduction	66φ	27φ	39φ	12φ	27φ	39φ	27φ	27φ	39φ
	20% reduction	72φ	30φ	42φ	12φ	30φ	42φ	30φ	30φ	42φ
	30% reduction	78φ	33φ	45φ	12φ	33φ	45φ	33φ	33φ	45φ
1Mbps mode 1Mbps	No reduction	24φ	10φ	14φ	4φ	10φ	14φ	10φ	10φ	14φ
	10% reduction	26φ	11φ	15φ	4φ	11φ	15φ	11φ	11φ	15φ
	20% reduction	29φ	13φ	16φ	4φ	13φ	16φ	12φ	13φ	16φ
	30% reduction	31φ	14φ	17φ	4φ	14φ	17φ	13φ	14φ	17φ

The above clock counts are values when HSCLK is selected for the operating frequency (I2M0CD[1:0]="000").
When 1/2 or 1/4HSCLK is selected, the counts increase in proportion to the dividing ratio.
Specify the I^2C operation clock as follows.

When the HSCLK is 24MHz
Standard mode : I^2C operation clock HSCLK~1/4HSCLK
Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK

φ: Clock cycle of 1/m HSCLK (m is selected by the I2m0CD[2:0] bits of the I2UM0MOD register)

Table 13-2 Relationship between Communication Speeds and HSCLK Clock Counts (HSCLK = 16 or 32 MHz)

Communication speed (I2M0SP)	Speed reduction (I2M0DW1, 0)	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Standard mode 100 kbps	No reduction	160φ	72φ	88φ	16φ	72φ	88φ	72φ	72φ	88φ
	10% reduction	176φ	80φ	96φ	16φ	80φ	96φ	80φ	80φ	96φ
	20% reduction	192φ	88φ	104φ	16φ	88φ	104φ	88φ	88φ	104φ
	30% reduction	208φ	96φ	112φ	16φ	96φ	112φ	96φ	96φ	112φ
Fast mode 400kbps	No reduction	40φ	14φ	26φ	12φ	14φ	26φ	14φ	14φ	26φ
	10% reduction	44φ	16φ	28φ	12φ	16φ	28φ	16φ	16φ	28φ
	20% reduction	48φ	18φ	30φ	12φ	18φ	30φ	18φ	18φ	30φ
	30% reduction	52φ	20φ	32φ	12φ	20φ	32φ	20φ	20φ	32φ
1Mbps mode 1Mbps	No reduction	16φ	6φ	10φ	4φ	6φ	10φ	6φ	6φ	10φ
	10% reduction	18φ	7φ	11φ	4φ	7φ	11φ	7φ	7φ	11φ
	20% reduction	19φ	8φ	11φ	4φ	8φ	11φ	7φ	8φ	11φ
	30% reduction	21φ	9φ	12φ	4φ	9φ	12φ	8φ	9φ	12φ

The above clock counts are values when HSCLK is selected for the operating frequency (I2M0CD[1:0]="000").
When 1/2 or 1/4HSCLK is selected, the counts increase in proportion to the dividing ratio.
Specify the I^2C operation clock as follows.

When the HSCLK is 16MHz
- Standard mode : I^2C operation clock HSCLK, 1/2HSCLK
- Fast mode : I^2C operation clock HSCLK
- 1Mbps mode : I^2C operation clock HSCLK

When the HSCLK is 32MHz
- Standard mode : I^2C operation clock HSCLK~1/4HSCLK
- Fast mode : I^2C operation clock HSCLK, 1/2HSCLK
- 1Mbps mode : I^2C operation clock HSCLK, 1/2HSCLK

φ: Clock cycle of 1/m HSCLK (m is selected by the I2m0CD[2:0] bits of the I2UM0MOD register)

[Note]
When the slave device uses the clock synchronization function (handshake function) which holds the I2CM0_SCL signal at a "L" level, the time t_{CYC} and time t_{LOW} are extended.

13.3.4 Pin Settings

I2CM0_SCL and I2CM0_SDA can be selected from multiple GPIOs.
Be sure to select one of the following combinations of GPIOs for I2CM0_SCL and I2CM0_SDA.

Function	I/O pin	Combination 1	Combination 2
I^2C Master	I2CM0_SDA	P06	P22
	I2CM0_SCL	P07	P23

GPIOs used for SDA and SCL must be set to input/output ports.

- When using the Pull-UP resistor built in the LSI (Output: N-channel open drain, Input: Pull-UP)

Byte symbol	PnMODm							
Bit symbol	Pn0MD3	Pn0MD2	Pn0MD1	Pn0MD0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
Setting value	0	0	1	1	1	0	1	1

n: Port number (n=0~3)
m: Bit number (n=0~7)

[Note]
Use external pull-up resistors for SDA pin and SCL pin refering to the I^2C bus specification. The internal pull-up resistors can not satisfy the I^2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.

Chapter 14 DMAC

14. DMAC (Direct Memory Access Controller)

14.1 General Description

ML62Q1000 series has two channels of DMA(Direct Memory Access) Controller, which transfers data between a peripheral circuit (SFR) and the built-in RAM without the CPU operation.

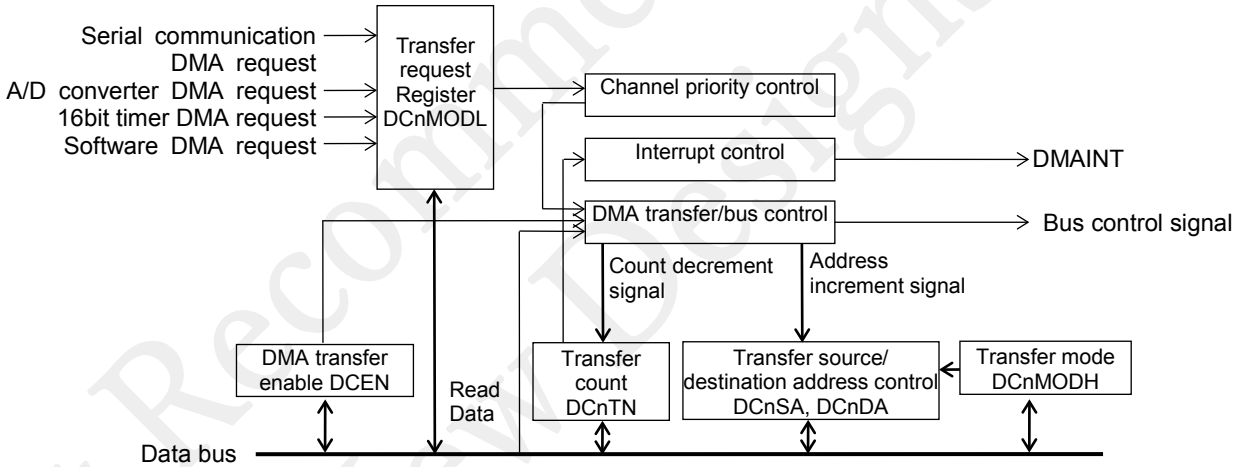
14.1.1 Features

The DMAC has the following features.

- Operation mode: Wait mode (Not available in No Wait mode)
- Transfer unit: 8bit/16bit
- Transfer count: 1 ~ 1024 time
- Transfer cycle: 2 cycle (CPU operation has priority if the access is competed)
- Transfer address: Fixed address / Increment address / Decrement address
- Transfer target: SFR/RAM ←→ SFR/RAM (Transfer from/to Flash is not supported)
- Transfer trigger: Serial communication DMA request, Successive approximation type A/D DMA request, 16bit timer DMA request and the software DMA request are seletable.
- Transfer priority: Channel 0 > Channel 1
- Interrupt: The DMAC interrupt occurs when the transfer at the last transfer count is completed.

14.1.2 Configuration

Figure 14-1 shows the configuration of the DMAC circuit.



DCnMODL : DMA channel n transfer mode register L
DCnMODH : DMA channel n transfer mode register H
DCnNT : DMA channel n transfer count register (word symbol)
DCnSA : DMA channel n transfer source address register (word symbol)
DCnDA : DMA channel n transfer destination address register (word symbol)
DCEN : DMA enable register
(n = 0,1)

Figure 14-1 Configuration of DMAC Circuit

14.2 Description of Registers

14.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF700	DMA channel 0 transfer mode register	DC0MODL	DC0MOD	R/W	8/16	0x00
0xF701		DC0MODH		R/W	8	0x00
0xF702	DMA channel 0 transfer count register	DC0TNL	DC0TN	R/W	8/16	0x00
0xF703		DC0TNH		R/W	8	0x00
0xF704	DMA channel 0 transfer source address register	DC0SAL	DC0SA	R/W	8/16	0x00
0xF705		DC0SAH		R/W	8	0x00
0xF706	DMA channel 0 transfer destination address register	DC0DAL	DC0DA	R/W	8/16	0x00
0xF707		DC0DAH		R/W	8	0x00
0xF708	DMA channel 1 transfer mode register	DC1MODL	DC1MOD	R/W	8/16	0x00
0xF709		DC1MODH		R/W	8	0x00
0xF70A	DMA channel 1 transfer count register	DC1TNL	DC1TN	R/W	8/16	0x00
0xF70B		DC1TNH		R/W	8	0x00
0xF70C	DMA channel 1 transfer source start address register	DC1SAL	DC1SA	R/W	8/16	0x00
0xF70D		DC1SAH		R/W	8	0x00
0xF70E	DMA channel 1 transfer destination start address register	DC1DAL	DC1DA	R/W	8/16	0x00
0xF70F		DC1DAH		R/W	8	0x00
0xF720	DMA transfer enable register	DCEN	-	R/W	8	0x00
0xF721	Reserved register	-	-	R	8	0x00
0xF722	DMA status register	DSTATL	DSTAT	R	8/16	0x00
0xF723		DSTATH		R	8	0x00
0xF724	DMA interrupt status clear register	DICLR	-	W	8	0x00
0xF725	Reserved register	-	-	W	8	0x00

[Note]
Registers with a word symbol are available for word access. For word access, specify an even address.

14.2.2 DMA Channel n Mode Register (DCnMOD: n = 0, 1)

Address: 0xF700, 0xF708
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	DCnMOD															
Byte symbol	DCnMODH								DCnMODL							
Bit symbol	DCnSTRG	-	-	DCnTRG4	DCnTRG3	DCnTRG2	DCnTRG1	DCnTRG0	-	-	-	DCnDS	DCnDAMD1	DCnDAMD0	DCnSAMD1	DCnSAMD0
Access type	W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	RW	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

DCnMOD (n = 0, 1) is a special function register (SFR) used to set the operation mode of channel n.
Set the transfer trigger, transfer unit, and addressing mode of the transfer source and transfer destination.

Description of bits

- **DCnSAMD1~0** (bits 1-0)
  The DCnSAMD1~0 bits are used to set the addressing mode of the transfer source of channel n.

DCnSAMD1	DCnSAMD0	Description
0	0	Fixed address mode (initial value)
0	1	Increment address mode Adds +1 for 8-bit transfer mode (DCnDS = 0) and +2 for 16-bit transfer mode (DCnDS = 1) to DCnSA.
1	0	Decrement address mode Adds -1 for 8-bit transfer mode (DCnDS = 0) and -2 for 16-bit transfer mode (DCnDS = 1) to DCnSA.
1	1	Do not use (Decrement address mode)

- **DCnDAMD1~0** (bits 3-2)
  The DCnDAMD1~0 bits are used to set the addressing mode of the transfer destination of channel n.

DCnDAMD1	DCnDAMD0	Description
0	0	Fixed address mode (initial value)
0	1	Increment address mode Adds +1 for 8-bit transfer mode (DCnDS = 0) and +2 for 16-bit transfer mode (DCnDS = 1) to DCnDA.
1	0	Decrement address mode Adds -1 for 8-bit transfer mode (DCnDS = 0) and -2 for 16-bit transfer mode (DCnDS = 1) to DCnDA.
1	1	Do not use (Decrement address mode)

- **DCnDS** (bit 4)
  The DCnDS bit is used to set the transfer unit of channel n.

DCnDS	Description
0	8 bits (initial value)
1	16 bits

- **DCnTRG4~0** (bits 12-8)
  The DCnTRG4~0 bits are used to select the DMA transfer trigger of channel n.

DCnTRG					Description
4	3	2	1	0	
0	0	0	0	0	No DMA request (initial value)
0	0	0	0	1	Successive approximation type A/D DMA request
0	0	0	1	0	Serial communication unit 00 DMA request
0	0	0	1	1	Serial communication unit 01 DMA request
0	0	1	0	0	Serial communication unit 10 DMA request
0	0	1	0	1	Serial communication unit 11 DMA request
0	0	1	1	0	Do not use
0	0	1	1	1	Do not use
0	1	0	0	0	Do not use
0	1	0	0	1	Do not use
0	1	0	1	0	Do not use
0	1	0	1	1	Do not use
0	1	1	0	0	Do not use
0	1	1	0	1	Do not use
0	1	1	1	0	Do not use
0	1	1	1	1	Do not use
1	0	0	0	0	16bit timer 0 DMA request
1	0	0	0	1	16bit timer 1 DMA request
1	0	0	1	0	16bit timer 2 DMA request t
1	0	0	1	1	16bit timer 3 DMA request
1	0	1	0	0	Do not use
1	0	1	0	1	Do not use
1	0	1	1	0	Do not use
1	0	1	1	1	Do not use
1	1	0	0	0	Do not use
1	1	0	0	1	Do not use
1	1	0	1	0	Do not use
1	1	0	1	1	Do not use
1	1	1	0	0	Do not use
1	1	1	0	1	Do not use
1	1	1	1	0	Do not use
1	1	1	1	1	Do not use

- **DCnSTRG** (bit 15)
  DCnSTRG is a write-only bit used to generates a software trigger of channel n. Only one transfer is performed by settng the DCnTRG bit to “1”. When it is read, "0" is always read.

DCnSTRG	Description
0	No function (initial)
1	Starts transfer once

[Note]
- Set the bits except for DCnSTRG bit when the transfer is disabled (DCnEN = 0).
- When performing the software trigger by setting the DCnSTRG bit to "1", the transfer is held if the next instruction is data memory access. Place two NOP instructions after setting DCnSTRG to "1" to prevent the hold and make the immediate transfer.
- When selecting the 16bit timer DMA request, select the 16bit timer mode by setting THn8BM bit of 16bit timer n mode register (TMHnMOD) to “0”.

14.2.3 DMA Channel n Transfer Count Register (DCnTN: n = 0, 1)

Address: 0xF702, 0xF70A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	DCnTN															
Byte symbol	DCnTNH								DCnTNL							
Bit symbol	-	-	-	-	-	-	DCnTN9	DCnTN8	DCnTN7	DCnTN6	DCnTN5	DCnTN4	DCnTN3	DCnTN2	DCnTN1	DCnTN0
Access type	R	R	R	R	R	R	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

DCnTN (n = 0, 1) is a special function register (SFR) used to set the transfer count for channel n.
The transfer count can be set to between 1 and 1024.
The transfer count is reduced by one each time DMA transfer is performed and when the transfer count is changed from 1 to 0, a DMAC interrupt (DMAINT) is generated.

Description of bits

- **DCnTN9 to DCnTN0** (bits 9-0)
  The DCnTN9 to DCnTN0 bits are used to set the transfer count for channel n.
  When the specified transfer count is completed, this register is set to 00H.

DCnTN										Transfer count
9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	1024 (initial value)
0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0	2
-	-	-	-	-	-	-	-	-	-	:
1	1	1	1	1	1	1	1	0	1	1021
1	1	1	1	1	1	1	1	1	0	1022
1	1	1	1	1	1	1	1	1	1	1023

[Note]
- Set the DCnTN when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).
- If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnTN is not guaranteed. Reconfigure the DCnTN when restarting the transfer.

14.2.4 DMA Channel n Transfer Source Address Register (DCnSA: n = 0, 1)

Address: 0xF704, 0xF70C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	DCnSA															
Byte symbol	DCnSAH								DCnSAL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Access type	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

DCnSA (n = 0, 1) is a special function register (SFR) used to set the transfer source address of channel n.
Specify an existing SFR address or RAM address. If a non-existing address is set, operation is not guaranteed.
In the increment/decrement address mode, an address is added or subtracted evey transfer. The address changes plus or minus 1(±1) in the 8-bit data transfer and plus or minus 2(±2) in the 16-bit data transfer. The lowest bit is ignored (d0=0) and even-numbered addresses are the targets in the 16-bit data transfer.

[Note]
- Set the DCnSA when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).
- If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnSA is not guaranteed. Reconfigure the DCnSA when restarting the transfer.

14.2.5 DMA Channel n Transfer Destination Address Register (DCnDA: n = 0, 1)

Address: 0xF706, 0xF70E
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	DCnDA															
Byte symbol	DCnDAH								DCnDAL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Access type	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

DCnDA (n = 0, 1) is a special function register (SFR) used to set the transfer destination address of channel n.
Specify an existing SFR address or RAM address. If a non-existing address is set, operation is not guaranteed.
In the increment/decrement address mode, an address is added or subtracted evey transfer. The address changes plus or minus 1(±1) in the 8-bit data transfer and plus or minus 2(±2) in the 16-bit data transfer. The lowest bit is ignored (d0=0) and even-numbered addresses are the targets in the 16-bit data transfer.

[Note]
- Set the DCnDA when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).
- If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnDA is not guaranteed. Reconfigure the DCnDA when restarting the transfer.

14.2.6 DMA Transfer Enable Register (DCEN)

Address: 0xF720
Access: R/W
Access size: 8 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								DCEN							
Bit symbol	-	-	-	-	-	-	-	-	DCF	-	-	-	-	-	DC1EN	DC0EN
Access type	R	R	R	R	R	R	R	R	R/W	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

DCEN is a special function register (SFR) used to enable the operation of the DMA transfer channel.

Description of bits

- **DC1EN to DC0EN** (bits 1-0)
  DCnEN (n = 1, 0) is used to enable the DMA transfer channel n.
  The DMA transfer starts when a transfer trigger is generated.
  When the transfer count set in the DMA channel transfer count register (DCnTN) is completed, it is automatically cleared to "0". Reset the DCnEN bit to “0” to stop the transfer.

	Description
0	Stop the DMA channel n transfer (initial value)
1	Enable the DMA channel n transfer

- **DCF** (bit 7)
  The DCF bit is used to fix the channel for until the transfer count set in the DMA channel transfer count register (DCnTN) is completed.
  When the DCF bit is set to "1", a transfer trigger of another channel is ignored until the current transfer is completed.
  If the DCnEN bit is enabled for all channels, one with a transfer trigger generated first starts the transfer. If a trigger is generated on both at the same time, channel 0 has the priority.

DCF	Description
0	DMA transfer channel is free (initial value)
1	DMA transfer channel is fixed

[Note]
- Set the DCF when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).
- When the specified transfer count is completed, the channel corresponding bit of the DMA interrupt status register (DSTATL) is set to "1". Be sure to clear the status bit before performing the next DMA transfer. When the status is set to "1", channel operation cannot be enabled. Clear the DCnISTA regardless using or not using the DMA interrupt.

14.2.7 DMA Status Register (DSTAT)

Address: 0xF722, 0xF723
Access: R
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	DSTAT															
Byte symbol	DCSTA								DCISTA							
Bit symbol	-	-	-	-	-	-	DC1STA	DC0STA	-	-	-	-	-	-	DC1ISTA	DC0ISTA
Access type	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

DSTAT is a special function register (SFR) used to read the status of the DMA transfer channel.
Set DCF to "1" to check if DMA transfer is being performed in the transfer channel fixed mode.
An interrupt is output from DMAC when the transfer count set in the DMA channel n transfer count register (DCTNn) is completed. Read DCISTA to check on which channel the interrupt occurred.

Description of bits

- **DCnISTA** (bits 1-0)
  DCnISTA (n = 1, 0) indicates that DMA transfer of channel n is completed.
  Check the transfer is finished after the interrupt occurred by reading the DCnISTA.
  The DCnISTA bit is cleared by writing "1" to corresponding bit of DMA interrupt status clear register (DICLRn).

DCnISTA	Description
0	DMA channel n is in operation or stopped (initial value)
1	Operation of DMA channel n is completed

- **DCnSTA** (bits 9-8)
  In the transfer channel fixed mode (DCF = “1”), DCnSTA (n = 1, 0) indicates that channel n is in transferring state. The bit is fixed to “0” in the transfer channel free mode (DCF = “0”).

DCnISTA	Description
0	DMA channel n is stopped (initial value)
1	DMA channel n is in transferring

14.2.8 DMA Interrupt Status Clear Register (DICLR)

Address: 0xF724
Access: W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								DICLR							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DICLR1	DICLR0
Access type	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

DICLR is a special function register (SFR) used to clear the interrupt status of the DMA transfer channel.

Description of bits

- **DICLR0** (bit 0)
  DC0ISTA bit of DSTATL register is cleared to "0" when writing "1" to DICLR0.
  When theDICLR0 bit is read, "0" is always read.

DICLR0	Description
0	Does not clear DMA channel 0 interrupt status
1	Clears DMA channel 0 interrupt status

- **DICLR1** (bit 1)
  DC1ISTA bit of DSTATL register is cleared to "0" by writing "1" to DICLR1.
  When the DICLR1 bit is read, "0" is always read.

DICLR1	Description
0	Does not clear the DMA channel 1 interrupt status
1	Clears the DMA channel 1 interrupt status

14.3 Description of Operation

14.3.1 DMA Operation Procedure

(1) Disable enable register (IE0~7) used as the transfer trigger, so that the trigger interrupt does not occur. If DMA communication is performed when the IE is ON, a trigger interrupt is generated.
(2) Set the transfer trigger, transfer unit, and addressing mode of the transfer source/destination of DCnMOD (DMA channel n mode register).
(3) Set the transfer count in DCnTN (DMA channel n transfer count register).
(4) Set the transfer source address in DCnSA (DMA channel n transfer source address register) and the transfer destination address in DCnDA (DMA channel n transfer destination address register).
(5) Set DCnEN of DCEN (DMA enable register) to "1" to put the LSI in the transfer trigger wait state. When a trigger is generated, transfer automatically starts.
(6) When the transfer count set in DCnTNn (DMA channel n transfer count register) is performed, a DMA interrupt (DMAINT) is generated.
(7) Read DCnISTA of DSTAT (DMA status register) to check which channel transfer is completed and write "1" to the bit corresponding bit of DICLRn to clear DCnISTAT.

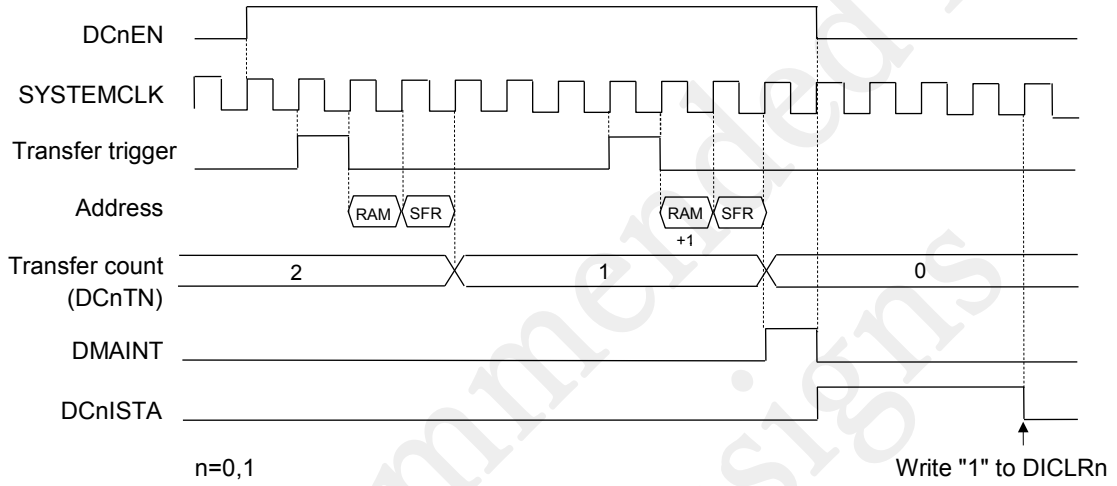


Figure 14-2 Channel n Transfer
(Transfer Source = RAM, Transfer Destination = SFR, Transfer Unit = 8 Bits, Transfer Count = 2)

[Note]
- The DMA transfer is held if the CPU continuously access the data memory, because the data memory access has a higher priority than the DMA transfer. If an interval is short, the transfer trigger may be overwritten. Take the transfer trigger interval time longer than that of the CPU continues the data memory access. Take Take four system clocks or longer in the transfer intervals in the case there is no data memory access.
- If a transfer trigger and a software trigger are generated at the same time, the transfer trigger is overwritten. Pay attention to a timing when a software trigger is generated.
- The DMA transfer is not available in HALT mode, HATL-H mode, STOP mode and STOP-D mode.
- Stop the clock for DMA by setting DCKDMA bit of BCKCON2 register, before entering into the HALT mode, HATL-H mode, STOP mode and STOP-D mode. Any trigger occurred in the standby mode is ignored. If restart to supply the clock by setting the DCKDMA bit to"0" the DMA transfer restarts. SFRs related to the DMA is not accessible when the clock for the DMA is stopped. Supply the clock before accessing the SFRs.
- Specify an area where the RAM and SFR exist for the transfer source/destination address. If a non-existing are is specified, "0" is transferred.
- Use the DMA transfer when the CPU mode is the wait mode.

14.3.2 UART Continuous Transmission

Figure 14-3 shows an example of flow chart for when making countious UART transmit by the DMA tranfer. See Chapter 11 “Serial Communication Unit“ for details about the UART.

[Example]

- Transfer 64 bytes continuously by using the DMA channel 0
- Use the serial communication unit 01 interrupt (SIU01INT) for the transfer trigger
- Transfer data in 0xE800 to 0xE840 of RAM to the SD0BUFH of the serial communication unit 0
- Transfer format: Full-duplex communication mode, 1200bps, 8bit length, with parity bit, one stop bit

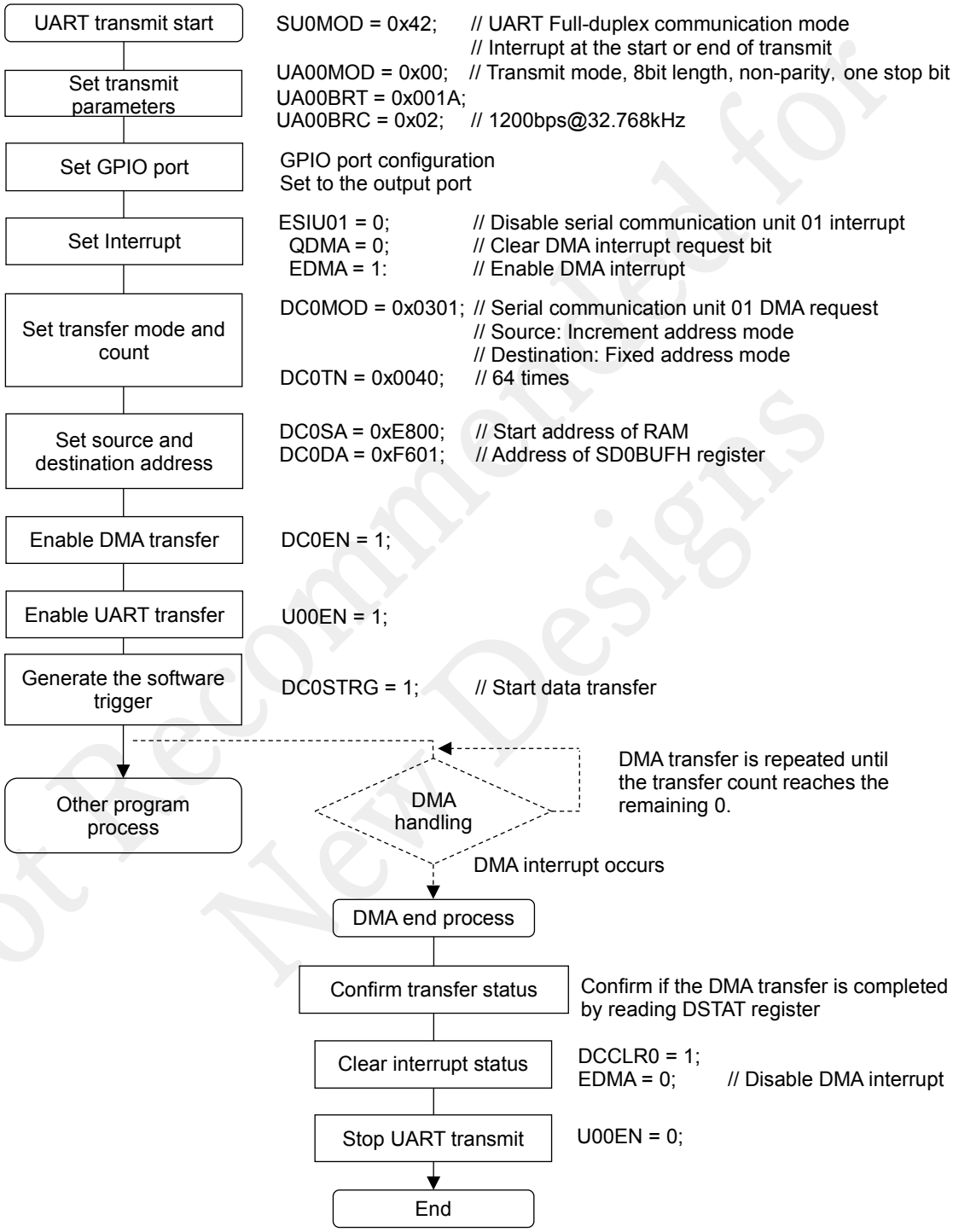


Figure 14-3 Flow chart of the countious UART transmit by the DMA tranfer

14.3.3 SSIO Continuous Transmission

Figure 14-4 shows an example of flow chart for when making countious SSIO transmit by the DMA tranfer. See Chapter 11 “Serial Communication Unit“ for details about the SSIO.

[Example]

- Transfer 64 bytes continuously by using the DMA channel 0
- Use the serial communication unit 00 interrupt (SIU00INT) for the transfer trigger
- Transfer data in 0xE800 to 0xE840 of RAM to the SD0BUFL of the serial communication unit 0
- Transfer format: SSIO mode, LSB first, transmit mode, 8bit length

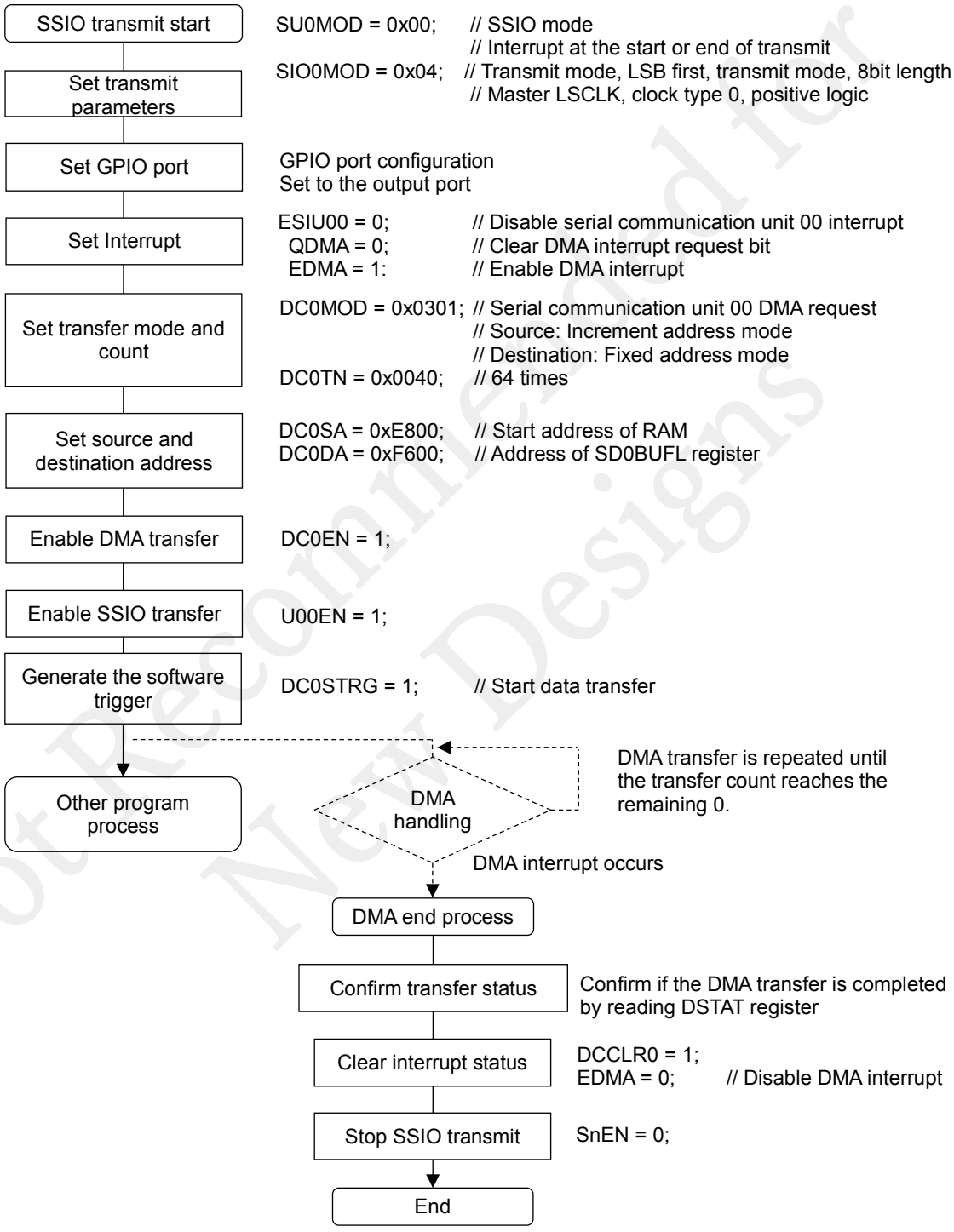


Figure 14-4 Flow chart of the countious SSIO transmit by the DMA tranfer

14.3.4 DMA Transfer Target Block

Table 14-1 shows the target function block that are useable as the DMA transfer source or destination.

Table 14-1 Useable functon block as the DMA transfer source or destination

	Function Block																									
	Memory				SFR (Specific Function Register)																					
	RAM	Program Flash	Code Option	Data Flash	Multiplier/Divider	Reset Function	Power Management	Interrupt	Clock Generation Circuit	Time Base Counter	16bit Timer	Functional Timer	Watch Dog Timer	Serial Communication Unit	I^2C Bus Unit	I^2C Bus Master	DMA Controller	Buzzer	General Port	External Interrupt	CRC Generator	Analog Comparator	D/A Converter	Voltage Level Supervisor (VLS)	SAR type A/D Converter	Flash Memory Control	Safety Function
Transfer Source	○	-	-	-	-	-	-	-	-	-	○	○	-	○	-	-	-	-	○	-	-	-	○	-	○	-	-
Transfer Destination	○	-	-	-	-	-	-	-	-	-	○	○	-	○	-	-	-	-	○	-	○	-	○	-	○	-	-

○: Useable -: Not useable

Chapter 15 Buzzer

15. Buzzer

15.1 General Description

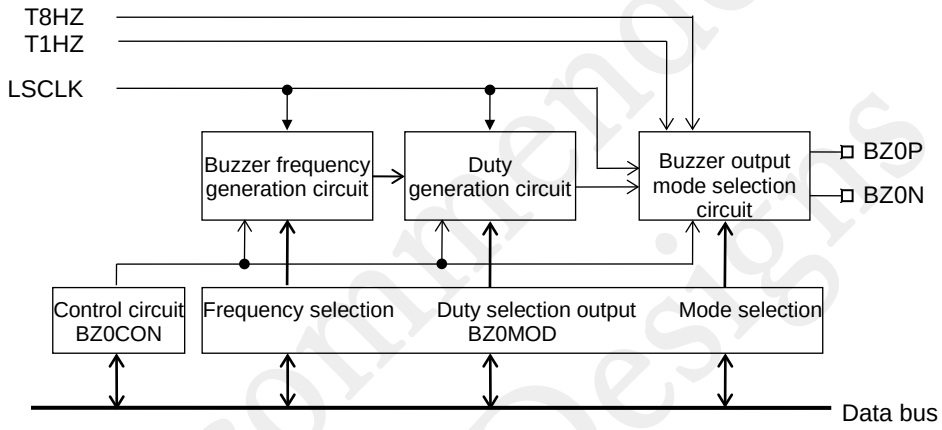
ML62Q1000 series has the buzzer circuit that generates a base signal in combination of eight frequencies and 15 duties and outputs it in four modes.
The buzzer has the positive phase output (BZ0P) and negative phase output (BZ0N). The buzzer output is assigned to the shared function of the GPIO. For details, see Chapter 17 "GPIO".
For the clock used in this block (T8HZ and T1HZ), see Chapter 7 "Low Speed Time Base Counter".

15.1.1 Features

- Four buzzer modes (Continuous sound/Single sound/Intermittent sound 1/Intermittent sound 2)
- Eight frequencies (4.096 kHz to 293 Hz)
- 15 duties (1/16 to 15/16)
  *However, only seven duties (1/8 to 7/8) are available when the buzzer frequency is 4.096 kHz.
- The initial level of the buzzer output pins can be selected

15.1.2 Configuration

Figure 15-1 shows the configuration of the buzzer circuit.



BZ0CON : Buzzer 0 control register
BZ0MOD : Buzzer 0 mode register

Figure 15-1 Configuration of Buzzer

15.1.3 List of Pins

The I/O pins of the buzzer signal are assigned to the secondary to octic functions of the GPIO.
For details about pin assignment and how to set the secondary to octic functions of the GPIO, see Chapter 17 "GPIO".

Pin name	I/O	Function
BZ0P	O	Buzzer 0 positive phase output
BZ0N	O	Buzzer 0 negative phase output

15.2 Description of Registers

15.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF0C0	Buzzer 0 control register	BZ0CON	-	R/W	8	0x00
0xF0C1	Reserved register	-	-	R	8	0x00
0xF0C2	Buzzer 0 mode register	BZ0MODL	BZ0MOD	R/W	8/16	0x00
0xF0C3		BZ0MODH		R/W	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

15.2.2 Buzzer 0 Control Register (BZ0CON)

Address: 0xF0C0
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	–															
Byte symbol	–								BZ0CON							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BZ0RUN
Access type	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

The buzzer 0 control register (BZ0CON) is a special function register (SFR) used to control the buzzer.

Description of bits

- **BZ0RUN** (bit 0)
  The BZ0RUN bit is used to start or stop the buzzer output.
  When the single sound output mode is selected by using the buzzer mode (BZ0MD1 to BZ0MD0) of the buzzer 0 mode register (BZ0MOD), if the buzzer output is automatically stopped, BZ0RUN is also automatically set to "0".

BZ0RUN	Description
0	Stops buzzer output (initial value)
1	Starts buzzer output

15.2.3 Buzzer 0 Mode Register (BZ0MOD)

Address: 0xF0C2
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	BZ0MOD															
Byte symbol	BZ0MODH								BZ0MODL							
Bit symbol	-	-	-	BZ0INI	BZ0D3	BZ0D2	BZ0D1	BZ0D0	-	BZ0F2	BZ0F1	BZ0F0	-	-	BZ0MD1	BZ0MD0
Access type	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	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

The buzzer 0 mode register (BZ0MOD) is a special function register (SFR) used to set the buzzer output waveform.

Description of bits

- **BZ0MD1 to BZ0MD0** (bits 1-0)
  The BZ0MD1 to BZ0MD0 bits are used to select the buzzer mode.
  For details of each buzzer, see Figure 15-3.

BZ0MD1	BZ0MD0	Description
0	0	Intermittent sound 1 output mode (initial value)
0	1	Intermittent sound 2 output mode
1	0	Single sound output mode
1	1	Continuous sound output mode

- **BZ0F2 to BZ0F0** (bits 6-4)
  The BZ0F2 to BZ0F0 bits are used to select the frequency of the buzzer output.

BZ0F2	BZ0F1	BZ0F0	Description
0	0	0	4.096kHz (initial value)
0	0	1	2.048kHz
0	1	0	1.024kHz
0	1	1	683Hz
1	0	0	512Hz
1	0	1	410Hz
1	1	0	341Hz
1	1	1	293Hz

- **BZ0D3 to BZ0D0** (bits 11-8)
  The BZ0D3 to BZ0D0 bits are used to select the duty of the buzzer output.

BZ0D3	BZ0D2	BZ0D1	BZ0D0	Description	
				Buzzer frequency At 4.096 kHz	Buzzer frequency Other than 4.096 kHz
0	0	0	0	1/8DUTY (initial value)	1/16DUTY (initial value)
0	0	0	1	1/8DUTY	1/16DUTY
0	0	1	0	1/8DUTY	2/16DUTY
0	0	1	1	1/8DUTY	3/16DUTY
0	1	0	0	2/8DUTY	4/16DUTY
0	1	0	1	2/8DUTY	5/16DUTY
0	1	1	0	3/8DUTY	6/16DUTY
0	1	1	1	3/8DUTY	7/16DUTY
1	0	0	0	4/8DUTY	8/16DUTY
1	0	0	1	4/8DUTY	9/16DUTY
1	0	1	0	5/8DUTY	10/16DUTY
1	0	1	1	5/8DUTY	11/16DUTY
1	1	0	0	6/8DUTY	12/16DUTY
1	1	0	1	6/8DUTY	13/16DUTY
1	1	1	0	7/8DUTY	14/16DUTY
1	1	1	1	7/8DUTY	15/16DUTY

- **BZ0INI** (bit 12)
  The BZ0INI bit is used to select the initial level of the buzzer output pins.

BZ0INI	Description
0	The initial level of the buzzer output pins (BZ0P/BZ0N) is "L" (initial value)
1	The initial level of the buzzer output pins (BZ0P/BZ0N) is "H"

15.3 Description of Operation

15.3.1 Operation Flow

Mak buzzer signal outputs, referring to the following procedure.
(1) Select the buzzer output format, frequency, duty, and initial level of the output pins in the buzzer 0 mode register (BZ0MOD).
(2) Set the GPIO as the heptic function and set it to the CMOS output mode.
(3) Write "1" to the BZ0RUN bit of the buzzer 0 control register (BZ0CON) to output the buzzer signal from the BZ0P and BZ0N pins.
Figure 15-2 shows an example of the operation flow of the buzzer output.

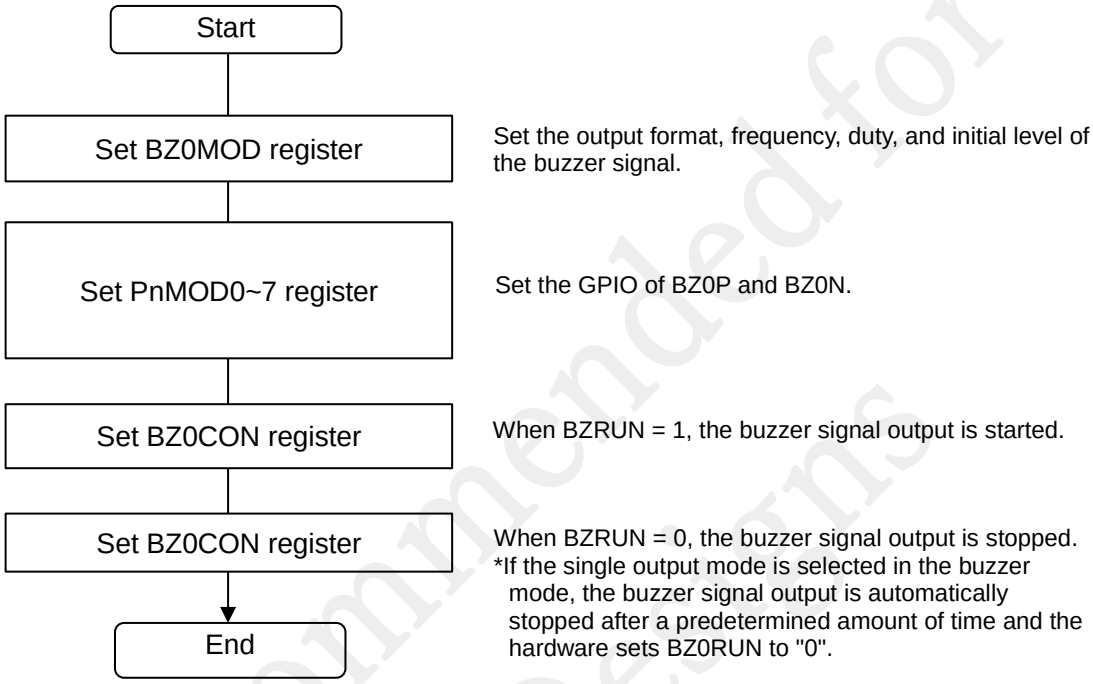


Figure 15-2 Example of Operation Flow of Buzzer Output

15.3.2 Buzzer Output Waveform

15.3.2.1 Buzzer Output Waveform

Figure 15-3 shows the buzzer output waveform of each buzzer mode.
An area painted in black indicates a section to which the buzzer signal pulse is output. For the detailed timing of the pulse signal, see Section 16.3.2.2.

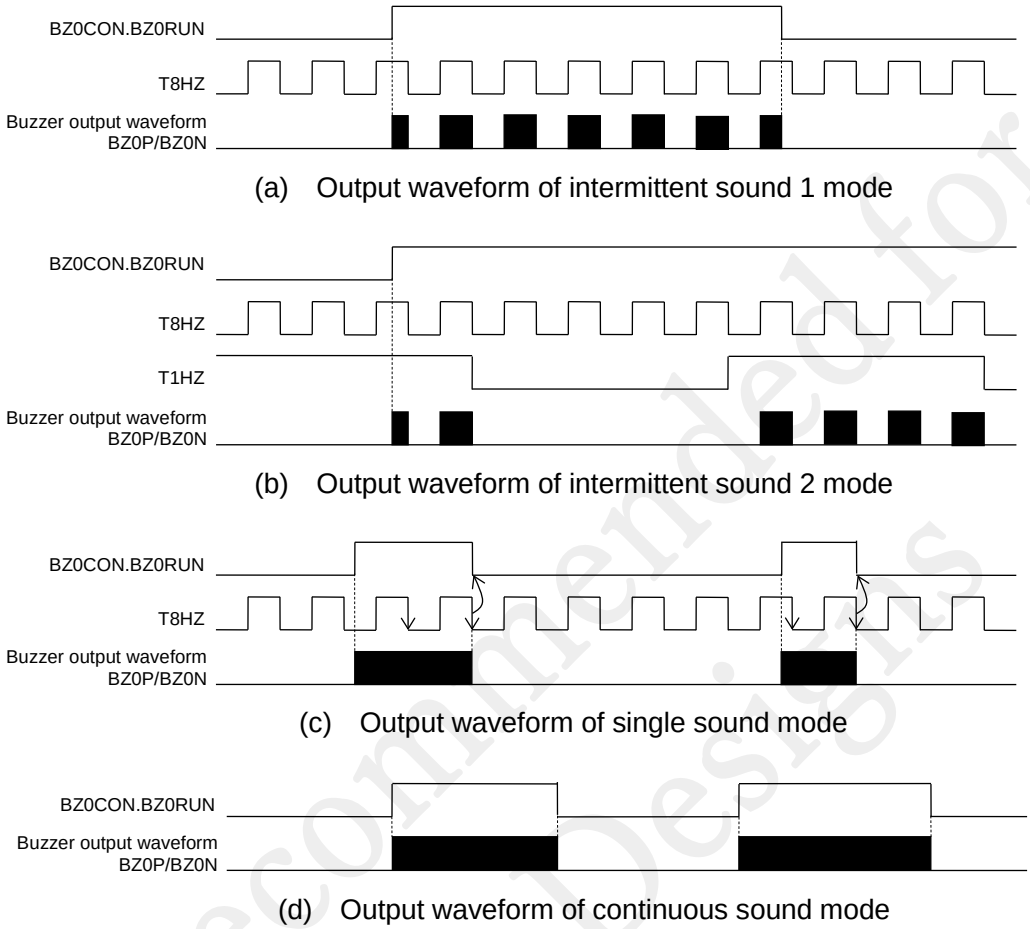
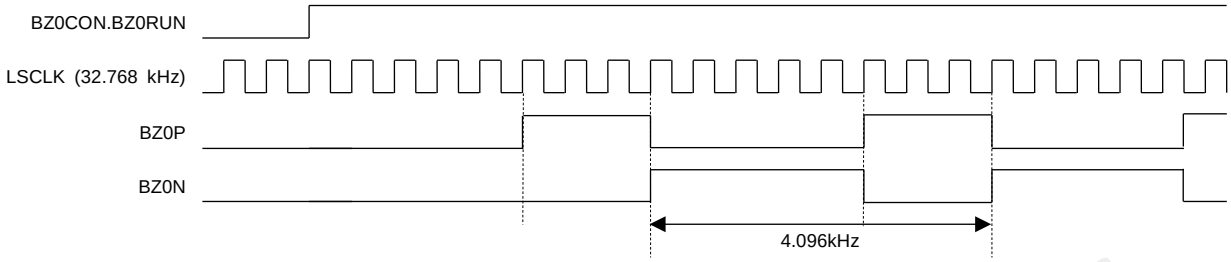


Figure 15-3 Output Waveform of Each Buzzer Mode

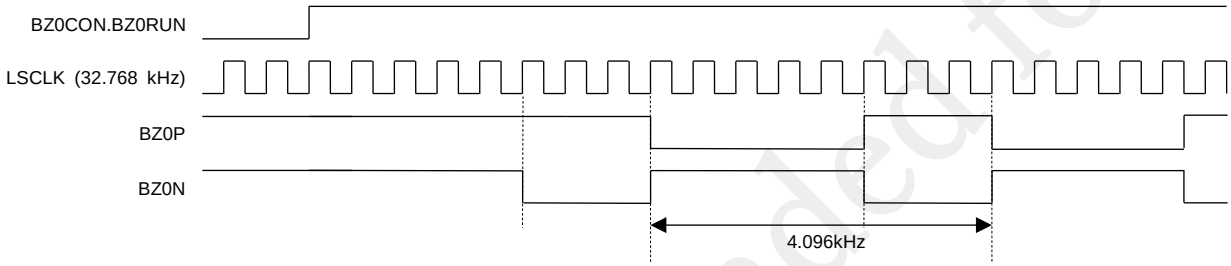
[Note]
When selecting the intermittent sound 1 mode, intermittent sound 2 mode or single sound mode, the buzzer output is not started from the first waveform dpending on the timing of BZ0RUN bit is set to “1”. To prevent it, set the BZ0RUN bit syncronizing at the falling edge of T8HZ or T1HZ signal by using the T8HZ/T1HZ time base counter interrupt or checking T8HZ/T1HZ bit of LTBR register.

15.3.2.2 Buzzer Output Start and Stop Timing

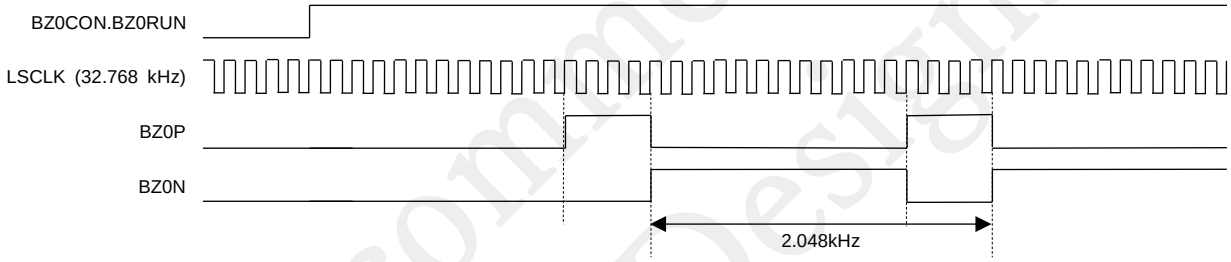
Figures 15-4 to 15-8 show the buzzer signal output start and stop timing waveform.



(1) When the initial value level is "L" (BZ0INI = 0)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty



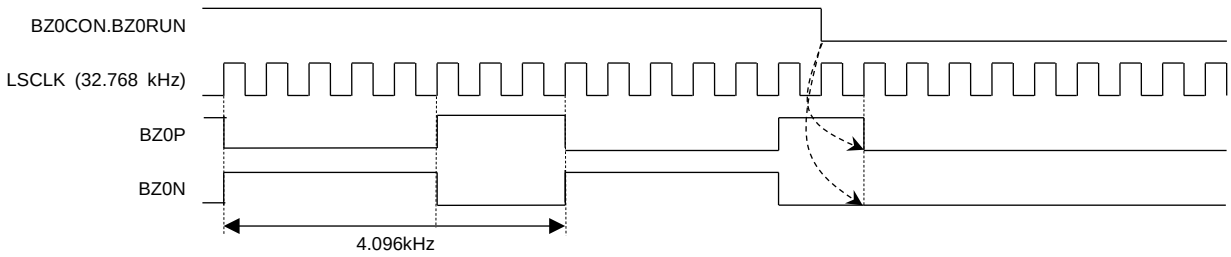
(2) When the initial value level is "H" (BZ0INI = 1)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty



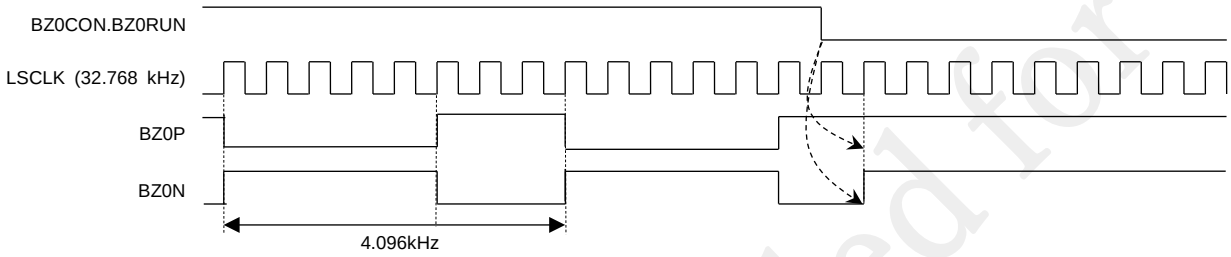
(3) When the initial value level is "L" (BZ0INI = 0)
Setting example of 2.048 kHz for the buzzer frequency and 4/16 for the duty

Figure 15-4 Buzzer Output Start Timing Controlled by BZ0RUN

[Note]
An error of up to LSCLK x 1 clock is generated before the buzzer output is started after BZ0RUN is set to "H".

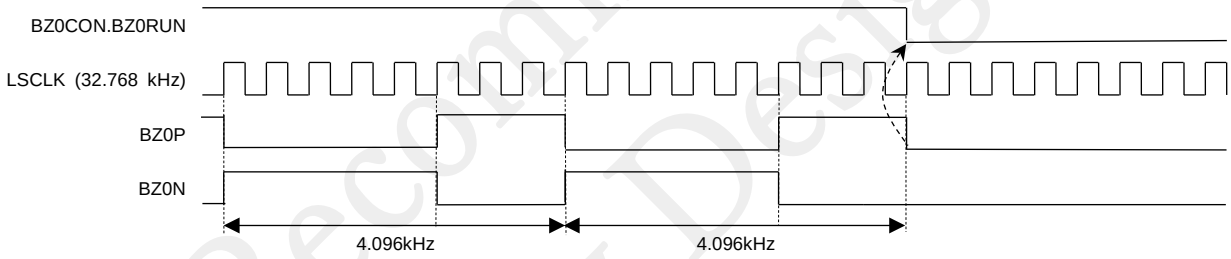


(1) When the initial value level is "L" (BZ0INI = 0)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

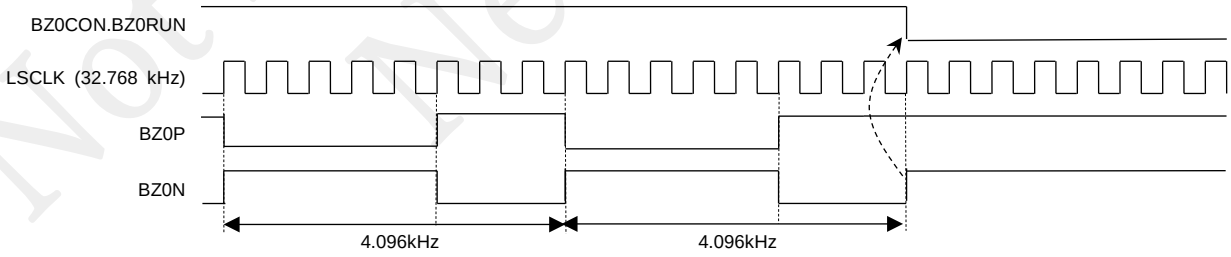


(2) When the initial value level is "H" (BZ0INI = 1)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

Figure 15-5 Buzzer Output Stop Timing Controlled by BZ0RUN

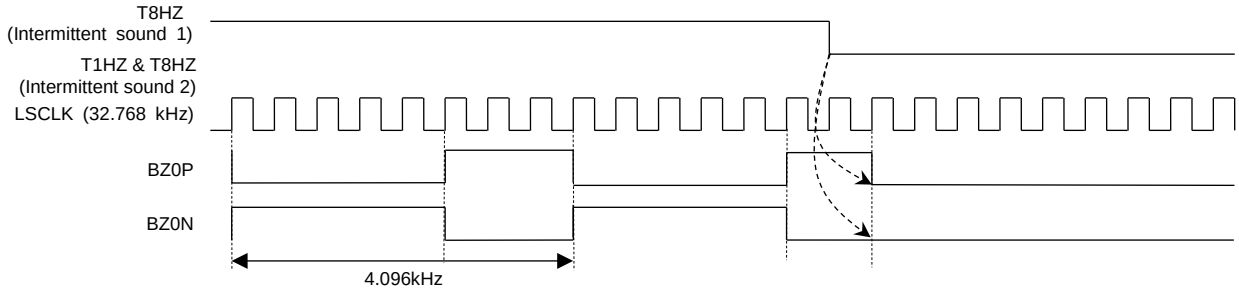


(1) When the initial value level is "L" (BZ0INI = 0)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

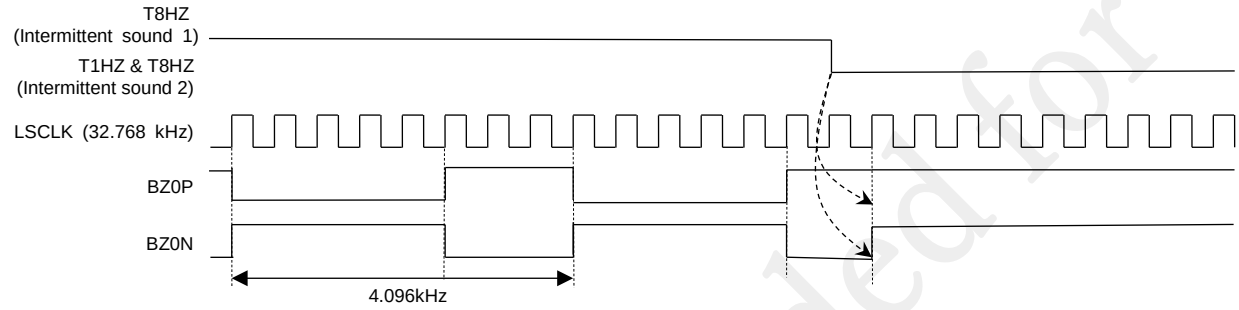


(2) When the initial value level is "H" (BZ0INI = 1)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

Figure 15-6 Buzzer Output Automatic Stop Timing in Single Sound Mode

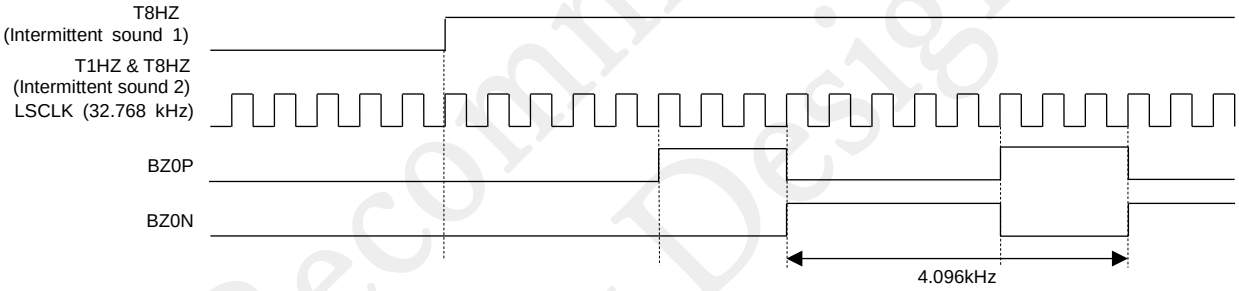


(1) When the initial value level is "L" (BZ0INI = 0)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

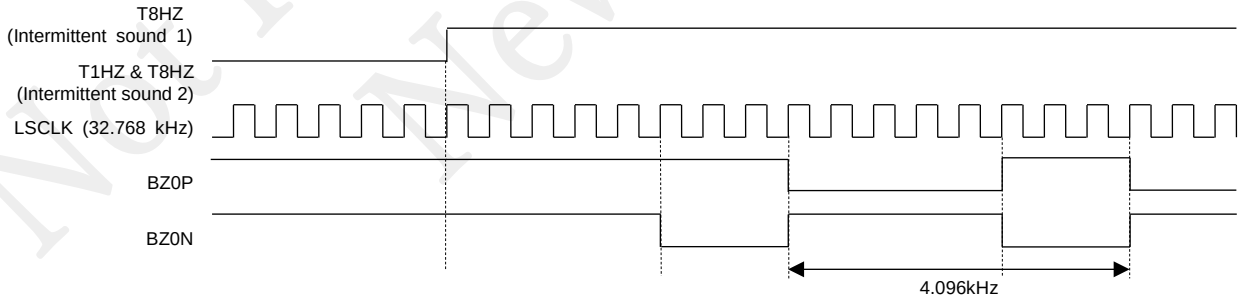


(2) When the initial value level is "H" (BZ0INI = 1)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

Figure 15-7 Buzzer Output Pause Timing in Intermittent Sound 1 Mode and Intermittent Sound 2 Mode



(1) When the initial value level is "L" (BZ0INI = 0)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty



(2) When the initial value level is "H" (BZ0INI = 1)
Setting example of 4.096 kHz for the buzzer frequency and 3/8 for the duty

Figure 15-8 Restart Timing from Buzzer Output Pause in Intermittent Sound 1 Mode or Intermittent Sound 2 Mode

[Note]
An error of up to LSCLK x 1 clock is generated before the buzzer output is started after T8HZ of the intermittent sound 1 mode or T8HZ & T1HZ of the intermittent sound 2 mode is set to "H".

15.3.3 Pin Settings

BZ0P and BZ0N can be selected from multiple GPIOs, but be sure to use one of the following combinations.

Function	Output pin	Combination 1	Combination 2
Buzzer	BZ0P	P26	P17
	BZ0N	P27	P20

Chapter 16 Simplified RTC

ML62Q1200A group does not have this function which is built-in other products of ML62Q1000 series

Chapter 17 GPIO

17. GPIO (General Purpose Input/Output)

17.1 General Description

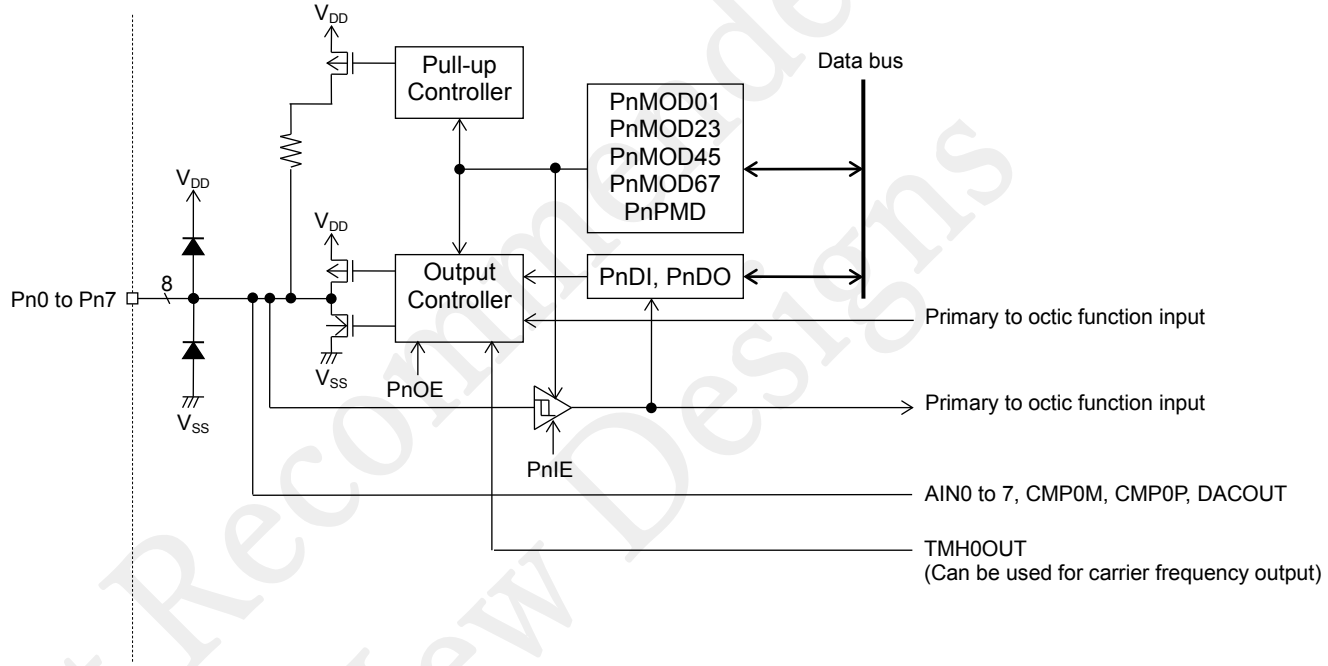
ML62Q1000 series has at most four general ports. Each port is configured with max. eight pins and input and output is switchable. The general ports can be used for external interrupts and external inputs for the functional timers. Also, used as input or output pins in shared functions by setting the port n mode register.
See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

17.1.1 Features

- Input or output is selectable
- Pull-up resister is selectable
- CMOS output or N-channel open drain output is selectable
- Direct driving LEDs is supported when the N-channel open drain is selected
- Carrier frequency output function
- Port output level test function

17.1.2 Configuration

Figure 17-1 shows the configuration of GPIOn.



PnDI : GPIOn data input register
PnDO : GPIOn data output register
PnMOD01 : GPIOn mode register
PnMOD23 : GPIOn mode register
PnMOD45 : GPIOn mode register
PnMOD67 : GPIOn mode register
PnPMD : GPIOn pulse mode register
n: 0~3 (PnMOD45 and PnMOD67 has only n=0~2)

Figure 17-1 Configuration of GPIOn

17.1.3 List of Pins

Table 17-1 List of Pins

Pin Name	Primary Function	Shared function *1							presence/ absence *2			
		Secondary function	Tertiary function	Quartic function	Quintic function	Sextic function	Heptic function	Octic function	16 pin	20 pin	24 pin	32 pin
P00	In/Out port	–	–	–	–	–	–	–	○	○	○	○
P01	In/Out port / DACOUT	–	–	–	–	–	–	–	—	—	○	○
P02	In/Out port / EXI0/ EXTRG0	SU0_RXD0/ SU0_SIN	–	–	FTM0P	OUTLSCLK	CMP0M	–	○	○	○	○
P03	In/Out port / EXI1/ EXTRG1	SU0_TXD0/ SU0_SOUT	SU0_TXD1	I2CU0_SDA	FTM0N	OUTHSCLK	CMP0P	–	○	○	○	○
P04	In/Out port / EXI2/ EXTRG2	SU0_SCLK	–	I2CU0_SCL	TMH0OUT	–	–	–	○	○	○	○
P05	In/Out port	–	–	–	–	–	–	–	—	○	○	○
P06	In/Out port	–	–	I2CM0_SDA	–	–	–	–	—	—	—	○
P07	In/Out port	SU0_RXD1	SU0_RXD0	I2CM0_SCL	–	–	–	–	—	—	—	○
P10	In/Out port	SU0_TXD1	–	–	–	–	–	–	—	—	—	○
P11	In/Out port	SU0_SCLK	–	–	–	–	–	–	—	—	—	○
P12	In/Out port	SU0_RXD0/ SU0_SIN	–	–	TMH4OUT	–	–	–	—	—	○	○
P13	In/Out port	SU0_TXD0/ SU0_SOUT	SU0_TXD1	–	TMH1OUT	–	TMH3OUT	–	○	○	○	○
P14	In/Out port	–	–	–	–	–	–	–	—	—	—	○
P15	In/Out port	–	–	I2CU0_SDA	–	–	–	–	—	—	—	○
P16	In/Out port	SU1_SCLK	–	I2CU0_SCL	TMH5OUT	–	–	–	—	—	○	○
P17	In/Out port /EXI3/ EXTRG3	SU0_RXD1	SU0_RXD0	–	FTM1P	–	BZ0P	AIN0	○	○	○	○
P20	In/Out port	SU0_TXD1	–	–	FTM1N	TBCOUT1	BZ0N	AIN1	○	○	○	○
P21	In/Out port /EXI4/ EXTRG4	SU1_RXD0/ SU1_SIN	–	–	FTM2P	OUTLSCLK	–	AIN2	○	○	○	○
P22	In/Out port	SU1_TXD0/ SU1_SOUT	SU1_TXD1	I2CM0_SDA	FTM2N	OUTHSCLK	–	AIN3	○	○	○	○
P23	In/Out port /EXI5/ EXTRG5	SU1_SCLK	–	I2CM0_SCL	TMH2OUT	–	–	V_{REF}	○	○	○	○
P24	In/Out port	SU1_RXD0/ SU1_SIN	–	–	–	–	–	AIN4	—	○	○	○
P25	In/Out port	SU1_TXD0/ SU1_SOUT	SU1_TXD1	–	–	–	–	AIN5	—	○	○	○
P26	In/Out port /EXI6/ EXTRG6	SU1_RXD1	SU1_RXD0	I2CU0_SDA	FTM3P	–	BZ0P	AIN6	○	○	○	○
P27	In/Out port / EXI7/ EXTRG7	SU1_TXD1	–	I2CU0_SCL	FTM3N	TBCOUT1	BZ0N	AIN7	○	○	○	○

Pin Name	Primary Function	Shared function *1							presence/ absence *2			
		Secondary function	Tertiary function	Quartic function	Quintic function	Sextic function	Heptic function	Octic function	16 pin	20 pin	24 pin	32 pin
P30	In/Out port	–	–	–	–	–	–	–	—	—	—	○
P31	In/Out port	–	–	–	–	–	–	–	—	—	—	○
P32	In/Out port	SU1_RXD1	SU1_RXD0	–	–	–	–	–	—	—	○	○
P33	In/Out port	SU1_TXD1	–	–	TMH3OUT	–	–	–	—	○	○	○

*1 If the non-existent functions indicated by "—" is configured as the output mode, "0" is output

*2 "○" indicates the port is present and "—" indicates the port is absent.

17.2 Description of Registers

17.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF200	Port 0 data register	P0DI	P0D	R/W	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	Port 0 mode register 01	P0MOD0	P0MOD01	R/W	8/16	0x01
0xF203		P0MOD1		R/W	8	0x00
0xF204	Port 0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x01
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	Port 0 pulse mode register	P0PMDL	P0PMD	R/W	8/16	0x00
0xF20B		P0PMDH		R/W	8	0x00
0xF20C to 0xF20F	Reserved register	-	-	R	8	0x00
0xF210	Port 1 data register	P1DI	P1D	R/W	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 to 0xF21F	Reserved register	-	-	R	8	0x00
0xF220	Port 2 data register	P2DI	P2D	R/W	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	Port 2 mode register 45	P2MOD4	P2MOD45	R/W	8/16	0x00
0xF227		P2MOD5		R/W	8	0x00
0xF228	Port 2 mode register 67	P2MOD6	P2MOD67	R/W	8/16	0x00
0xF229		P2MOD7		R/W	8	0x00
0xF22A	Port 2 pulse mode register	P2PMDL	P2PMD	R/W	8/16	0x00
0xF22B		P2PMDH		R/W	8	0x00

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF22C to 0xF22F	Reserved register	-	-	R	8	0x00
0xF230	Port 3 data register	P3DI	-	R/W	8	0x00
0xF231		P3DO	-	R/W	8	0x00
0xF232	Port 3 mode register 01	P3MOD0	P3MOD01	R/W	8/16	0x00
0xF233		P3MOD1		R/W	8	0xFF
0xF234	Port 3 mode register 23	P3MOD2	P3MOD23	R/W	8/16	0x00
0xF235		P3MOD3		R/W	8	0x00
0xF236	Reserved register	-	-	R	8	0x00
0xF237		-	-	R	8	0x00
0xF238		-	-	R	8	0x00
0xF239		-	-	R	8	0x00
0xF23A	Port 3 pulse mode register	P3PMDL	P3PMD	R/W	8/16	0x00
0xF23B		P3PMDH		R/W	8	0x00
0xF23C to 0xF23F	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

Table 17-2 List of Registers/Bits

Port Name	Pin Name	Control register / bit *1					presence/absence *2			
		Port data output register (PnDO)	Port data input register (PnDI)	Port mode register (PnMODm)	Port pulse mode register H (PnPMDH)	Port pulse mode register L (PnPMDL)	16 pin	20 pin	24 pin	32 pin
Port 0	P00	P00DO	P00DI	P0MOD0	—	—	○	○	○	○
	P01	P01DO	P01DI	P0MOD1	—	—	—	—	○	○
	P02	P02DO	P02DI	P0MOD2	—	—	○	○	○	○
	P03	P03DO	P03DI	P0MOD3	P03PLVL	P03PEN	○	○	○	○
	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	P11PLVL	P11PEN	—	—	—	○
	P12	P12DO	P12DI	P1MOD2	—	—	—	—	○	○
	P13	P13DO	P13DI	P1MOD3	P13PLVL	P13PEN	○	○	○	○
	P14	P14DO	P14DI	P1MOD4	—	—	—	—	—	○
	P15	P15DO	P15DI	P1MOD5	—	—	—	—	—	○
	P16	P16DO	P16DI	P1MOD6	—	—	—	—	○	○
	P17	P17DO	P17DI	P1MOD7	—	—	○	○	○	○
Port 2	P20	P20DO	P20DI	P2MOD0	P20PLVL	P20PEN	○	○	○	○
	P21	P21DO	P21DI	P2MOD1	—	—	○	○	○	○
	P22	P22DO	P22DI	P2MOD2	P22PLVL	P22PEN	○	○	○	○
	P23	P23DO	P23DI	P2MOD3	—	—	○	○	○	○
	P24	P24DO	P24DI	P2MOD4	—	—	—	○	○	○
	P25	P25DO	P25DI	P2MOD5	P25PLVL	P25PEN	—	○	○	○
	P26	P26DO	P26DI	P2MOD6	—	—	○	○	○	○
	P27	P27DO	P27DI	P2MOD7	P27PLVL	P27PEN	○	○	○	○
Port 3	P30	P30DO	P30DI	P3MOD0	—	—	—	—	—	○
	P31	P31DO	P31DI	P3MOD1	—	—	—	—	—	○
	P32	P32DO	P32DI	P3MOD2	—	—	—	—	○	○
	P33	P33DO	P33DI	P3MOD3	P33PLVL	P33PEN	—	○	○	○

*1 Corresponding bits and registers for each pin.
*2 "○" indicates the port is present and "—" indicates the port is absent.

17.2.2 Port n Data Register (PnD) n: Port Number (0~3)

Address: 0xF2n0
Access: R
Access size: 8/16 bits
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnD															
Byte symbol	PnDO								PnDI							
Bit symbol	Pn7DO	Pn6DO	Pn5DO	Pn4DO	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7DI	Pn6DI	Pn5DI	Pn4DI	Pn3DI	Pn2DI	Pn1DI	Pn0DI
Access type	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

PnD is a special function register (SFR) used to read the level of the port n pin and write output data.
The input level of the port n pin can be read by reading PnDI in the input mode.
Data written to PnDO in the output mode are output to the port n pin. The PnDO is readable. Enable or disable the input or output by using the port n mode register.

- **Pn7DI to Pn0DI** (bits 7-0)

Setting value	Description
0	The input level is "L" (initial value)
1	The input level is "H"

- **Pn7DO to Pn0DO** (bits 15-8)

Setting value	Description
0	Outputs "L" (initial value)
1	Outputs "H"

17.2.3 Port n Mode Register 01 (PnMOD01) n: Port Number (0~3)

Address: 0xF2n2
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnMOD01															
Byte symbol	PnMOD1								PnMOD0							
Bit symbol	Pn1MD3	Pn1MD2	Pn1MD1	Pn1MD0	Pn1OD	Pn1PU	Pn1OE	Pn1IE	Pn0MD3	Pn0MD2	Pn0MD1	Pn0MD0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
Access type	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	*

*The initial value of P00IE and P00PU for the Port0 is "1" and other bits are "0".

PnMOD01 is a special function register (SFR) to select the input/output mode, input/output status, and primary to octic function of Pn0 pin and Pn1 pin.

- **Pn0IE** (bit 0)
  The Pn0IE bit is used to enable the input from Pn0 pin.

Pn0IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn0OE** (bit 1)
  The Pn0OE bit is used to enable the output to Pn0 pin.

Pn0OE	Description
0	Disable the output (initial value)
0	Enable the output

- **Pn0PU** (bit 2)
  The Pn0PU bit is used to have the internal pull-up resistor to the Pn0 pin. The Pn0PU bit is valid when the port has the condtions of "input is enabled and output is disabled", "input is enabled and output is enabled and Nch open drain output is selected".

Pn0PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn0PU valid/invalid condtion

Pn0OE	Pn0IE	Pn0OD	Description
1	0	*	Pn0PU is valid
1	1	1	
Others			Pn0PU is invalid

*0 or 1(don't care)

- **Pn0OD** (bit 3)
  The Pn0OD bit is used to select the output type of Pn0 pin.
  LED is 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 avility.

Pn0OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn0MD3 to Pn0MD0** (bits 7-4)
  The Pn0MD3 to Pn0MD0 bits are used to select the primary to octic function of Pn0 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn0MD3	Pn0MD2	Pn0MD1	Pn0MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Do not use (Primary function)

*0 or 1(don't care)

- **Pn1IE** (bit 8)
  Pn1IE bit is used to enable the input of Pn1 pin.

Pn1IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn1OE** (bit 9)
  The Pn1OE bit is used to enable the output of Pn1 pin.

Pn1OE	Description
0	Disable the output (initial value)
1	Enable the output

- **Pn1PU** (bit 10)
  The Pn1PU bit is used to have the internal pull-up resistor to the Pn1 pin. The Pn1PU bit is valid when the port has the condtions of "input is enabled and output is disabled", "input is enabled and output is enabled and Nch open drain output is selected".

Pn1PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn1PU valid/invalid condtion

Pn1OE	Pn1IE	Pn1OD	Description
1	0	*	Pn1PU is valid

1	1	1	
Others			Pn1PU is invalid

*0 or 1(don't care)

- **Pn1OD** (bit 11)
  The Pn1OD bit is used to select the output type of Pn1 pin.
  LED is 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 avility.

Pn1OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn1MD3 to Pn1MD0** (bits 15-12)
  The Pn0MD3 to Pn0MD0 bits are used to select the primary to octic function of Pn0 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn1MD3	Pn1MD2	Pn1MD1	Pn1MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Primary function

*0 or 1(don't care)

[Note]
Set the port n mode register before setting the external interrupt registers (EICON0 and EIMOD0) and the interrupt enable register (IE1), If setting the port n mode register when the interrupt is enabled, unexpected may happen.

17.2.4 Port n Mode Register 23 (PnMOD23) n: Port Number (0~3)

Address: 0xF2n4
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnMOD23															
Byte symbol	PnMOD3								PnMOD2							
Bit symbol	Pn3MD3	Pn3MD2	Pn3MD1	Pn3MD0	Pn3OD	Pn3PU	Pn3OE	Pn3IE	Pn2MD3	Pn2MD2	Pn2MD1	Pn2MD0	Pn2OD	Pn2PU	Pn2OE	Pn2IE
Access type	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

PnMOD23 is a special function register (SFR) to select the input/output mode, input/output status, and primary to octic function of Pn2 pin and Pn2 pin.

- **Pn2IE** (bit 0)
  The Pn2IE bit is used to enable the input from Pn2 pin.

Pn2IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn2OE** (bit 1)
  The Pn2OE bit is used to enable the output to Pn2 pin.

Pn2OE	Description
0	Disable the output (initial value)
0	Enable the output

- **Pn2PU** (bit 2)
  The Pn2PU bit is used to have the internal pull-up resistor to the Pn2 pin. The Pn2PU bit is valid when the port has the condtions of "input is enabled and output is disabled", "input is enabled and output is enabled and Nch open drain output is selected".

Pn2PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn2PU valid/invalid condtion

Pn2OE	Pn2IE	Pn2OD	Description
1	0	*	Pn2PU is valid
1	1	1	
Others			Pn2PU is invalid

*0 or 1(don't care)

- **Pn2OD** (bit 3)
  The Pn2OD bit is used to select the output type of Pn2 pin.
  LED is 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 avility.

Pn2OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn2MD3 to Pn2MD0** (bits 7-4)
  The Pn2MD3 to Pn2MD0 bits are used to select the primary to octic function of Pn2 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn2MD3	Pn2MD2	Pn2MD1	Pn2MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Do not use (Primary function)

*0 or 1(don't care)

- **Pn3IE** (bit 8)
  Pn3IE bit is used to enable the input of Pn3 pin.

Pn3IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn3OE** (bit 9)
  The Pn3OE bit is used to enable the output of Pn3 pin.

Pn3OE	Description
0	Disable the output (initial value)
1	Enable the output

- **Pn3PU** (bit 10)
The Pn3PU bit is used to have the internal pull-up resistor to the Pn3 pin. The Pn3PU bit is valid when the port has the condtions of “input is enabled and output is disabled”, “input is enabled and output is enabled and Nch open drain output is selected”.

Pn3PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn3PU valid/invalid condtion

Pn3OE	Pn3IE	Pn3OD	Description
1	0	*	Pn3PU is valid
1	1	1	
Others			Pn3PU is invalid

*0 or 1(don’t care)

- **Pn3OD** (bit 11)
The Pn3OD bit is used to select the output type of Pn3 pin.
LED is 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 avility.

Pn3OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn3MD3 to Pn3MD0** (bits 15-12)
The Pn0MD3 to Pn0MD0 bits are used to select the primary to octic function of Pn0 pin.
For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn3MD3	Pn3MD2	Pn3MD1	Pn3MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Primary function

*0 or 1(don’t care)

[Note]
Set the port n mode register before setting the external interrupt registers (EICON0 and EIMOD0) and the interrupt enable register (IE1), If setting the port n mode register when the interrupt is enabled, unexpected interrupts may happen.

17.2.5 Port n Mode Register 45 (PnMOD45) n: Port Number (0~2)

Address: 0xF2n6
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnMOD45															
Byte symbol	PnMOD5								PnMOD4							
Bit symbol	Pn5MD3	Pn5MD2	Pn5MD1	Pn5MD0	Pn5OD	Pn5PU	Pn5OE	Pn5IE	Pn4MD3	Pn4MD2	Pn4MD1	Pn4MD0	Pn4OD	Pn4PU	Pn4OE	Pn4IE
Access type	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

PnMOD45 is a special function register (SFR) to select the input/output mode, input/output status, and primary to octic function of Pn4 pin and Pn5 pin.

- **Pn4IE** (bit 0)
  The Pn4IE bit is used to enable the input from Pn4 pin.

Pn4IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn4OE** (bit 1)
  The Pn4OE bit is used to enable the output to Pn4 pin.

Pn4OE	Description
0	Disable the output (initial value)
0	Enable the output

- **Pn4PU** (bit 2)
  The Pn4PU bit is used to have the internal pull-up resistor to the Pn4 pin. The Pn4PU bit is valid when the port has the condtions of "input is enabled and output is disabled", "input is enabled and output is enabled and Nch open drain output is selected".

Pn4PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn4PU valid/invalid condtion

Pn4OE	Pn4IE	Pn4OD	Description
1	0	*	Pn4PU is valid
1	1	1	
Others			Pn4PU is invalid

*0 or 1(don't care)

- **Pn4OD** (bit 3)
  The Pn4OD bit is used to select the output type of Pn4 pin.
  LED is 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 avility.

Pn4OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn4MD3 to Pn4MD0** (bits 7-4)
  The Pn4MD3 to Pn4MD0 bits are used to select the primary to octic function of Pn4 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn4MD3	Pn4MD2	Pn4MD1	Pn4MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Do not use (Primary function)

*0 or 1(don't care)

- **Pn5IE** (bit 8)
  Pn5IE bit is used to enable the input of Pn5 pin.

Pn5IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn5OE** (bit 9)
  The Pn5OE bit is used to enable the output of Pn5 pin.

Pn5OE	Description
0	Disable the output (initial value)
1	Enable the output

- **Pn5PU** (bit 10)
  The Pn5PU bit is used to have the internal pull-up resistor to the Pn5 pin. The Pn5PU bit is valid when the port has the condtions of “input is enabled and output is disabled”, “input is enabled and output is enabled and Nch open drain output is selected”.

Pn5PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn5PU valid/invalid condtion

Pn5OE	Pn5IE	Pn5OD	Description
1	0	*	Pn5PU is valid
1	1	1	
Others			Pn5PU is invalid

*0 or 1(don’t care)

- **Pn5OD** (bit 11)
  The Pn5OD bit is used to select the output type of Pn5 pin.
  LED is 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 avility.

Pn5OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn5MD3 to Pn5MD0** (bits 15-12)
  The Pn0MD3 to Pn0MD0 bits are used to select the primary to octic function of Pn0 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn5MD3	Pn5MD2	Pn5MD1	Pn5MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Primary function

*0 or 1(don’t care)

[Note]
Set the port n mode register before setting the external interrupt registers (EICON0 and EIMOD0) and the interrupt enable register (IE1), If setting the port n mode register when the interrupt is enabled, unexpected may happen.

17.2.6 Port n Mode Register 67 (PnMOD67) n: Port Number (0~2)

Address: 0xF2n8
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnMOD67															
Byte symbol	PnMOD7								PnMOD6							
Bit symbol	Pn7MD3	Pn7MD2	Pn7MD1	Pn7MD0	Pn7OD	Pn7PU	Pn7OE	Pn7IE	Pn6MD3	Pn6MD2	Pn6MD1	Pn6MD0	Pn6OD	Pn6PU	Pn6OE	Pn6IE
Access type	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

PnMOD67 is a special function register (SFR) to select the input/output mode, input/output status, and primary to octic function of Pn6 pin and Pn7 pin.

- **Pn6IE** (bit 0)
  The Pn6IE bit is used to enable the input from Pn6 pin.

Pn6IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn6OE** (bit 1)
  The Pn6OE bit is used to enable the output to Pn6 pin.

Pn6OE	Description
0	Disable the output (initial value)
0	Enable the output

- **Pn6PU** (bit 2)
  The Pn6PU bit is used to have the internal pull-up resistor to the Pn6 pin. The Pn6PU bit is valid when the port has the condtions of "input is enabled and output is disabled", "input is enabled and output is enabled and Nch open drain output is selected".

Pn6PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn6PU valid/invalid condtion

Pn6OE	Pn6IE	Pn6OD	Description
1	0	*	Pn6PU is valid
1	1	1	
Others			Pn6PU is invalid

*0 or 1(don't care)

- **Pn6OD** (bit 3)
  The Pn6OD bit is used to select the output type of Pn6 pin.
  LED is 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 avility.

Pn6OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn6MD3 to Pn6MD0** (bits 7-4)
  The Pn6MD3 to Pn6MD0 bits are used to select the primary to octic function of Pn6 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn6MD3	Pn6MD2	Pn6MD1	Pn6MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Do not use (Primary function)

*0 or 1(don't care)

- **Pn7IE** (bit 8)
  Pn7IE bit is used to enable the input of Pn7 pin.

Pn7IE	Description
0	Disable the input (initial value)
1	Enable the input

- **Pn7OE** (bit 9)
  The Pn7OE bit is used to enable the output of Pn7 pin.

Pn7OE	Description
0	Disable the output (initial value)
1	Enable the output

- **Pn7PU** (bit 10)
  The Pn7PU bit is used to have the internal pull-up resistor to the Pn7 pin. The Pn7PU bit is valid when the port has the condtions of “input is enabled and output is disabled”, “input is enabled and output is enabled and Nch open drain output is selected”.

Pn7PU	Description
0	Without a pull-up resistor (initial value)
1	With a pull-up resistor

Pn7PU valid/invalid condtion

Pn7OE	Pn7IE	Pn7OD	Description
1	0	*	Pn7PU is valid
1	1	1	
Others			Pn7PU is invalid

*0 or 1(don’t care)

- **Pn7OD** (bit 11)
  The Pn7OD bit is used to select the output type of Pn7 pin.
  LED is 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 avility.

Pn7OD	Description
0	CMOS output (initial value)
1	N-channel open drain output

- **Pn7MD3 to Pn7MD0** (bits 15-12)
  The Pn0MD3 to Pn0MD0 bits are used to select the primary to octic function of Pn0 pin.
  For the details of the primary to octic functions, see Table 17-1 "List of Pins".

Setting value				Description
Pn7MD3	Pn7MD2	Pn7MD1	Pn7MD0	
0	0	0	0	Primary function (initial value)
0	0	0	1	Secondary function
0	0	1	0	Tertiary function
0	0	1	1	Quartic function
0	1	0	0	Quintic function
0	1	0	1	Sextic function
0	1	1	0	Heptic function
0	1	1	1	Octic function
1	*	*	*	Primary function

*0 or 1(don’t care)

[Note]
Set the port n mode register before setting the external interrupt registers (EICON0 and EIMOD0) and the interrupt enable register (IE1), If setting the port n mode register when the interrupt is enabled, unexpected may happen.

17.2.7 Port n Pulse Mode Register (PnPMD) n: Port Number (0~3)

Address: 0xF2nA
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	PnPMD															
Byte symbol	PnPMDH								PnPMDL							
Bit symbol	Pn7PLVL	Pn6PLVL	Pn5PLVL	Pn4PLVL	Pn3PLVL	Pn2PLVL	Pn1PLVL	Pn0PLVL	Pn7PEN	Pn6PEN	Pn5PEN	Pn4PEN	Pn3PEN	Pn2PEN	Pn1PEN	Pn0PEN
Access type	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

PnPMD is a special function register (SFR) used when outputing a carrier frequency to the port n.
See Table 17-2 “List of Registers/Bits” for details about the corresponding pins and bits. Set “0” to non-existing bits.

- **Pn7PEN to Pn0PEN** (bits 7-0)
  The Pn7PEN to Pn0PEN bits are used to enable or disable the carrier frequency output to Pn7~Pn0 pin. These bits are valid when the Pn7~Pn0 pins are configured as the output is enabled (Pn7OE~Pn0OE = “0”).

PnmPEN	Description
0	Disable the carrier frequency output (initial value)
1	Enable the carrier frequency output

m: bit No.

- **Pn7PLVL to Pn0PLVL** (bits 15-8)
  The Pn7PLVL to Pn0PLVL bits are used to select the condition of syncronizing level for outputting the carrier frequency (Output of 16-bit timer 0/TMH0OUT) to the pins.

PnmPLVL	Description
0	Outputs the carrier frequency to the pins when the output level is "H" (initial value)
1	Outputs the carrier frequency to the pins when the output level is "L"

m: bit No.

17.3 Description of Operation

17.3.1 Input Function

For each pin of Port n (n: port number 0~3), specify the PnmIE and PnmOE bits (m: bit number 0~7) of the Port n mode register and enable the inputs.
In the input mode, the input level of each pin of Port n can be read from the Port n data input register (PnDI). Also, the pull-up resisotr is configuarable by setting the PnmPU bit.
At a system reset, the input is disabled and no pull-up is selected as the initial setting.

17.3.2 Output Function

For each pin of Port n (n: port number 0~3), specify the PnmOD bit of the Port n mode register to select either of the N-channel open drain output mode or CMOS output mode and specify the PnmOE bit to enable the output.
In the output mode, "L" or "H" level is output to each pin of Port n, depending on the value set by the Port n data output register (PnDO).
At a system reset, the output is disabled and CMOS output mode is selected as the initial setting.

17.3.3 Primary Function Other Than Input/Output Function

External interrupt inputs (EXI0~EXI7) and functional timer external inputs (EXTRG0~EXTRG7) are available as the primary function. Enable the input (Set PnmIE bit to “1”) when using the external interrupt inputs and the functional timer external inputs.
For more details about the external interrupt inputs and the functional timer external inputs, see Chapter 17 “GPIO Interrupt Control Circuit” and Chapter 9 “Functional Timer”.
Set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to “0” as “Disable the input” and “Disable the output”.

17.3.4 Shared Function (Secondary to Octic Function)

For each pin of Port n (n: port number 0~3), secondary to octic functions are configurable.
Select a function by setting the PnmMD3 ~ PnmMD0 of the Port n mode register and specify the input/output conditions depending on the functions. See Table “17-1 List of Pins” for details on the shared functions.

17.3.5 Carrier Frequency Output Function

A carrier frequency signal can be output from the port n by setting Port n Pulse Mode (PnPMD) Register.
See Table 17-2 “List of Registers/Bits” for the applicable pins for the carrier frequency output.
TMH0OUT signal of 16-bit timer 0 is used for the carrier frequency output. For details about the 16-bit timer, see Chapter 8 "16-bit Timer".
Figures 17-2 and 17-3 show an example when outputing the carrier frequency signal.

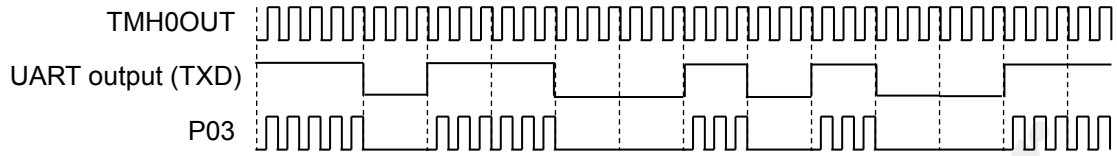


Figure 17-2 Example of Timing When P03 is Assigned to UART Output Pin
(P03PEN = "1", P03PLVL = "0")

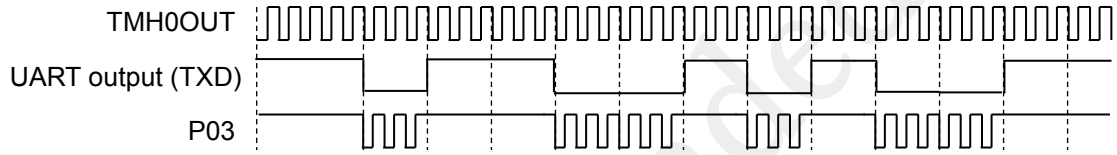


Figure 17-3 Example of Timing When P03 is Assigned to UART Output Pin
(P03PEN = "1", P03PLVL = "1")

17.3.6 GPIO Test Function

When setting PnmOE bit and PnmIE bit of the Port n mode register to “1”, output level specified in the port n data output register (PnDO) is readable from the port n data input register (PnDI), which is available to internally confirm the output level is surely driven to the port.

17.3.7 Setting Example of Port

Figure 17-4 shows an example of setting P07-P00 as the CMOS output mode to output 0x55.
Set the port n data output register at first, then enable the output by setting the port n mode register, so that unexpected level is not output from the port.

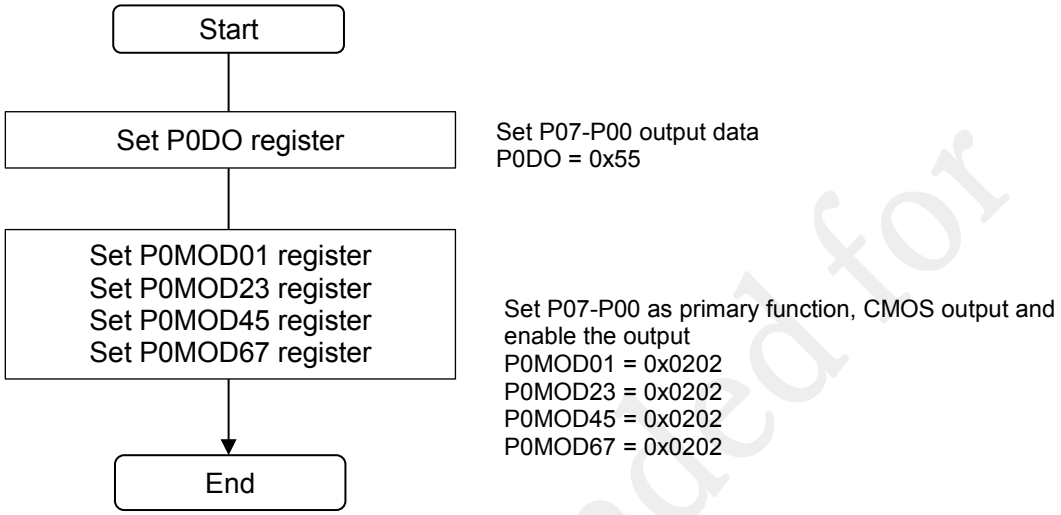


Figure 17-4 Setting Example of Port 0

17.3.8 Notes for using the P00/TEST0 pin

P00/TEST0 pin is used for the general port, the on-chip debug function or ISP funcction.
Confirm following notes in each usage.
Also, the P00/TEST0 is initially configured as the input with pull-up register. If input “L” level at the initial setting, the input current flows.

17.3.8.1 When using as the general purpose port

Make sure following notes when using the reset function by the RESET_N pin.

- Set the P00/TEST0 pin to “H” level input or “pull-up” when the RESET_N pin gets to “H” level. or
- Hold the P00/TEST0 pin to “H” level input or “pull-up” 1ms before and after the RESET_N pin gets to “H” level.

Figure 17-5 shows the sequence.
See Chapter 3 “Reset Function” for details about the RESET_N pin reset.

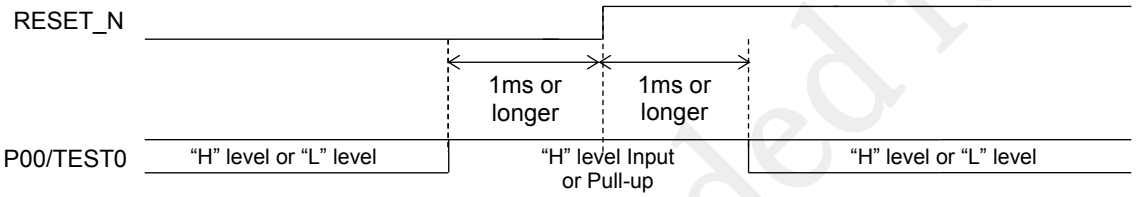


Figure 17-5 P00/TEST0 pin setting when releasing the RESET_N pin to “H” level

17.3.8.2 When using the On-chip debug function and ISP function

When using the on-chip debug function or ISP function, P00/TEST0 cannot be used as the general purpose port.

- Do not program the software that makes the P00/TEST0 pin output mode.
- Do not connect external components onto the P00/TEST0 pin.

See Chapter 28 “On-Chip Debug Function” for details about the On-chip Debug function and see Chapter 25.4 “ISP function” for details about the ISP function.

Chapter 18 External Interrupt Control Circuit

18. External Interrupt Control Circuit

18.1 General Description

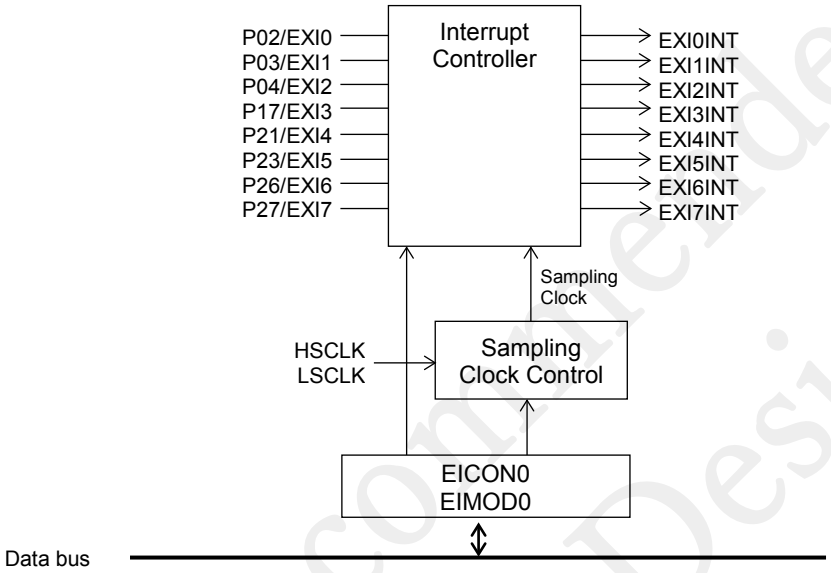
ML62Q1000 series has the external interrupt function that generates interrupts by signals input to the general ports. The interrupt channel has each dedicated interrupt vector. For details about the interrupt vector, see Chapter 5 "Interrupt".

18.1.1 Features

- Maskable interrupt function
- Allows selection of the interrupt disabled mode, falling-edge interrupt mode, rising-edge interrupt mode, or both-edge interrupt mode
- Allows selection of with/without interrupt sampling (the sampling clock is LSCLK or HSCLK)

18.1.2 Configuration

Figure 18-1 shows the configuration of the external interrupt control.



EICON0 : External interrupt control register 0
EIMOD0 : External interrupt mode register 0

Figure 18-1 Configuration of External Interrupt Control

18.1.3 List of Pins

The external interrupt is assigned to the primary function of the general port.
For details of pin assignment, refer to Chapter 17 "GPIO".

Pin name	I/O	Description
P02/EXI0	I	External interrupt 0 pin
P03/EXI1	I	External interrupt 1 pin
P04/EXI2	I	External interrupt 2 pin
P17/EXI3	I	External interrupt 3 pin
P21/EXI4	I	External interrupt 4 pin
P23/EXI5	I	External interrupt 5 pin
P26/EXI6	I	External interrupt 6 pin
P27/EXI7	I	External interrupt 7 pin

18.2 Description of Registers

18.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF044	External interrupt control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved register	-	-	R	8	0x00
0xF047	Reserved register	-	-	R	8	0x00
0xF048	External interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A	Reserved register	-	-	R	8	0x00
0xF04B	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

18.2.2 External Interrupt Control Register 01 (EICON0)

Address: 0xF044
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	EICON0															
Byte symbol	EICON0H								EICON0L							
Bit symbol	PI7E1	PI6E1	PI5E1	PI4E1	PI3E1	PI2E1	PI1E1	PI0E1	PI7E0	PI6E0	PI5E0	PI4E0	PI3E0	PI2E0	PI1E0	PI0E0
Access type	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

EICON0 is a special function register (SFR) used to select the interrupt edge of the external interrupt.

Description of setting values

Setting value		Description
PInE1	PInE0	
0	0	Interrupt disabled (initial value)
0	1	Falling-edge interrupt
1	0	Rising-edge interrupt
1	1	Both-edge interrupt

Supported blocks

Bit	Bit symbol name		Supported interrupt
Bit 15, 7	PI7E1	PI7E0	EXI7INT interrupt (P27/EXI7)
Bit 14, 6	PI6E1	PI6E0	EXI6INT interrupt (P26/EXI6)
Bit 13, 5	PI5E1	PI5E0	EXI5INT interrupt (P23/EXI5)
Bit 12, 4	PI4E1	PI4E0	EXI4INT interrupt (P21/EXI4)
Bit 11, 3	PI3E1	PI3E0	EXI3INT interrupt (P17/EXI3)
Bit 10, 2	PI2E1	PI2E0	EXI2INT interrupt (P04/EXI2)
Bit 9, 1	PI1E1	PI1E0	EXI1INT interrupt (P03/EXI1)
Bit 8, 0	PI0E1	PI0E0	EXI0INT interrupt (P02/EXI0)

18.2.3 External Interrupt Mode Register 0 (EIMOD0)

Address: 0xF048
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	EIMOD0															
Byte symbol	EIMOD0H								EIMOD0L							
Bit symbol	-	PG0DIV2	PG0DIV1	PG0DIV0	-	PG0CS0	-	-	PI7SM	PI6SM	PI5SM	PI4SM	PI3SM	PI2SM	PI1SM	PI0SM
Access type	R	R/W	R/W	R/W	R	R/W	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

EIMOD0 is a special function register (SFR) to select the sampling clock and with/without sampling for the External interrupt. Only one sampling clock can be selected in EXI0 to EXI7.

- **PI7SM to PI0SM** (bits 7-0)
  The PI7SM to PI0SM bits are used to select with/without sampling for each external interrupt.

  Description of setting values

Setting value	Description
0	Detects the input signal edge for interrupt without sampling (initial value)
1	Detects with sampling

  Supported blocks

Bit	Bit symbol name	Supported interrupt
Bit 7	PI7SM	EXI7INT interrupt (P27/EXI7)
Bit 6	PI6SM	EXI6INT interrupt (P26/EXI6)
Bit 5	PI5SM	EXI5INT interrupt (P23/EXI5)
Bit 4	PI4SM	EXI4INT interrupt (P21/EXI4)
Bit 3	PI3SM	EXI3INT interrupt (P17/EXI3)
Bit 2	PI2SM	EXI2INT interrupt (P04/EXI2)
Bit 1	PI1SM	EXI1INT interrupt (P03/EXI1)
Bit 0	PI0SM	EXI0INT interrupt (P02/EXI0)

- **PG0CS0** (bit 10)
  The PG0CS0 bit is used to select the sampling clock.
  One sampling clock can be selected in the EXI0 to EXI7 group.

PG0CS0	Description
0	Sampling with LSCLK (initial value)
1	Sampling with HSCLK

- **PG0DIV2 to PG0DIV0** (bits 14-12)
  The PG0DIV2 to PG0DIV0 bits are used to select the dividing ratio of the sampling clock for the external interrupt.
  One sampling clock can be selected in the EXI0 to EXI7 group.

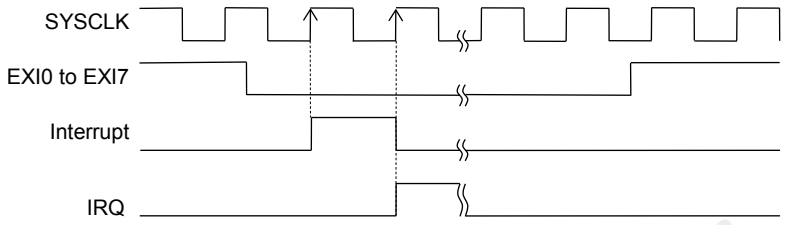
PG0DIV2	PG0DIV1	PG0DIV0	Description
0	0	0	1/1 (initial value)
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	1/64
1	1	1	1/1

[Note]
- In the STOP/STOP-D/HALT-H mode, no sampling is performed regardless of the values set in PI7SM to PI0SM since the sampling clock stops.
  In this switching from "with sampling" to "without sampling", there is a time period (*) in which interrupts gets disabled.
  When entering to STOP/STOP-D/HALT-H mode, specify the external interrupt as "without sampling".
  After returning from STOP/STOP-D/HALT-H mode, specify the PI7SM to PI0SM as "with sampling" if needed.
  (*) When entering to STOP/STOP-D mode: Max.30us
  When returing from STOP/STOP-D mode: Max.250us(When LSCLK is selected for CPU),
  Max.2.5ms(When HSCLK is selected for CPU)
  When returing from HALT-H mode: Max.2.5ms(When HSCLK is selected for CPU)
- When the HSCLK is selected and ENOSC bit of FCON register is "0", the sampling function is not available.
- If HSCLK is selected for the sampling block when the high-speed clock is not oscillating, the sampling circuit does not operate. If sampling is performed with HSCLK, enable the high-speed clock oscillation in advance. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".

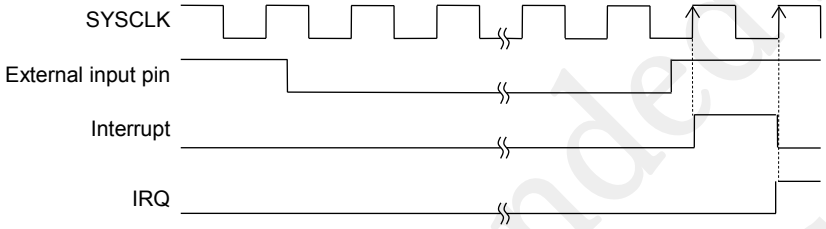
18.3 Description of Operation

18.3.1 Interrupt Request Timing

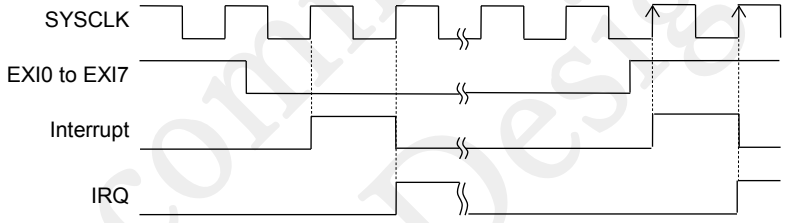
Figure 18-2 shows the interrupt generation timing in rising-edge interrupt mode, in falling-edge interrupt mode, and in both-edge interrupt mode without sampling, and in rising-edge interrupt mode with sampling.



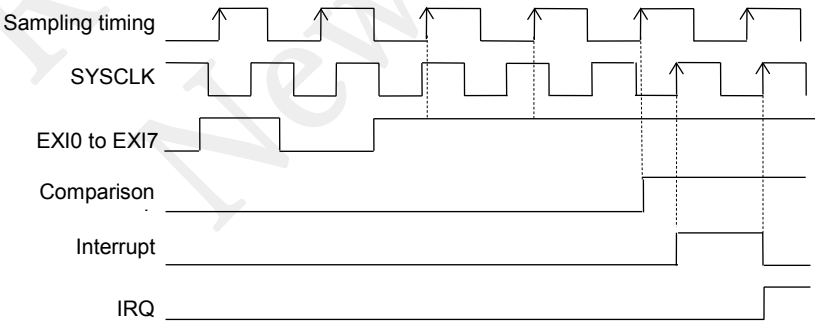
(a) When falling-edge interrupt mode without sampling is selected



(b) When rising-edge interrupt mode without sampling is selected



(c) When both-edge interrupt mode without sampling is selected



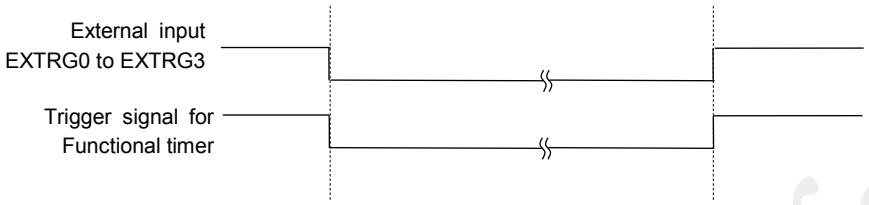
(d) When rising-edge interrupt mode with sampling is selected

Figure 18-2 External Interrupt Generation Timing

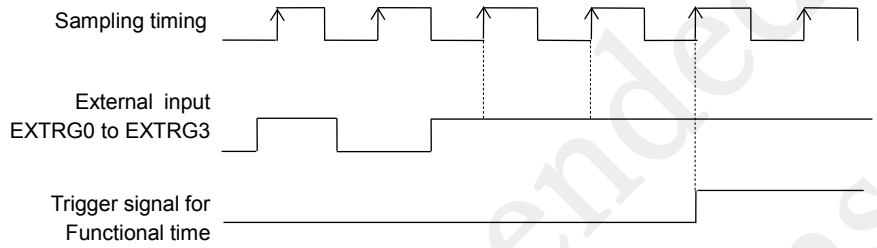
18.3.2 External Trigger Signal to Functional Timer

Some of pins assigned as External interrupt can be used as an external trigger (EXTRG0 to EXTRG3) of the function timer. In this case, the sampling function of the control circuit for external interrupt can be used.

Figure 18-3 shows the timing of a signal output to the function timer.



(a) Without sampling



(b) With sampling

Figure 18-3 Function timer trigger signal

18.3.3 External Interrupt Setting Flow

Figure 18-4 shows the setting flow of External interrupt.

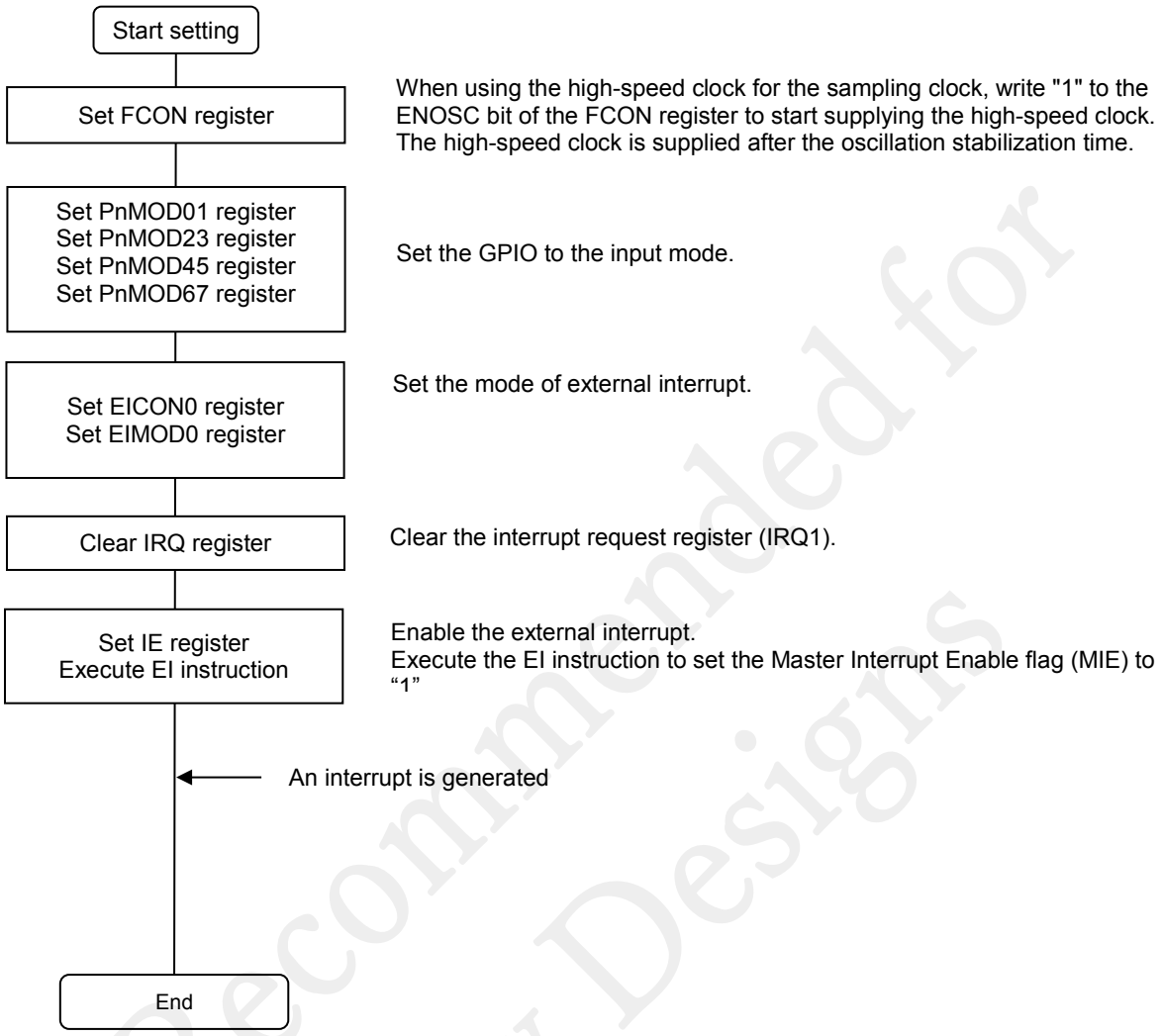


Figure 18-4 External Interrupt Setting Flow

Chapter 19 CRC Generator

19. CRC (Cycle Redundancy Check) Generator

19.1 General Description

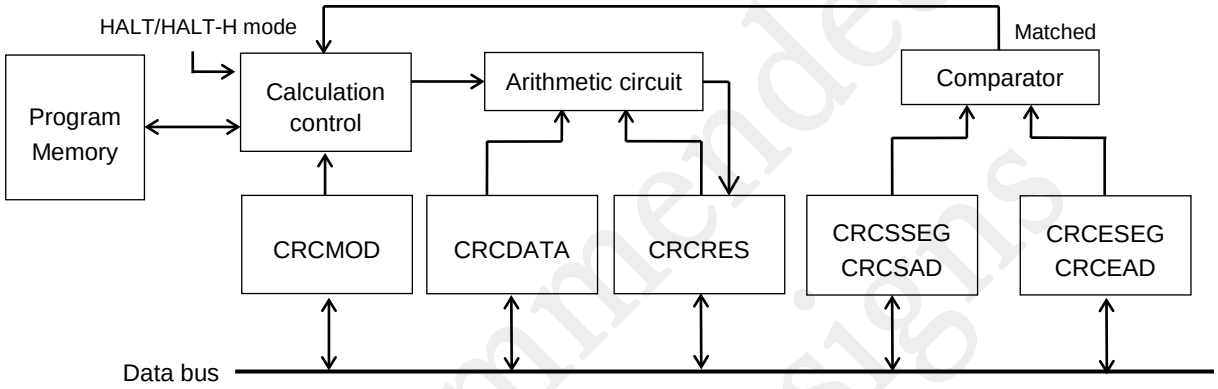
ML62Q1000 series has the CRC (Cycle Redundancy Check) generator that performs CRC calculation and generates the CRC data used for error detection in serial communications. Also, the CRC generator has automatic CRC calculation mode to check data in program memory, which is available in HALT mode or HALT-H mode.

19.1.1 Features

- Generates CRC data from data set in CRC calculation register by the software (Calculation unit is 8bit)
- Automatic CRC calculation to check data in program memory in HALT or HALT-H mode and generates CRC data (Calculation unit is 32bit)
- LSB first
- Generator polynomial: $x^{16} + x^{12} + x^5 + 1$

19.1.2 Configuration

Figure 19-1 shows the configuration of the CRC.



CRCMOD : CRC mode register
CRCDATA : CRC data register
CRCRES : CRC calculation result register
CRCSSEG : Automatic CRC calculation start segment setting register
CRCSAD : Automatic CRC calculation start address setting register
CRCESEG : Automatic CRC calculation end segment setting register
CRCEAD : Automatic CRC calculation end address setting register

Figure 19-1 Configuration of CRC

19.2 Description of Registers

19.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF0D0	Automatic CRC calculation start address setting register	CRCSADL	CRCSAD	R/W	8/16	0x00
0xF0D1		CRCSADH		R/W	8	0x00
0xF0D2	Automatic CRC calculation end address setting register	CRCEADL	CRCEAD	R/W	8/16	0xFC
0xF0D3		CRCEADH		R/W	8	0xFF
0xF0D4	Automatic CRC calculation start segment setting register	CRCSSEG	-	R/W	8	0x00
0xF0D5	Reserved register	-	-	R	8	0x00
0xF0D6	Automatic CRC calculation end segment setting register	CRCESEG	-	R/W	8	0x0F
0xF0D7	Reserved register	-	-	R	8	0x00
0xF0D8	CRC data register	CRCDATA	-	R/W	8	0x00
0xF0D9	Reserved register	-	-	-	8	0x00
0xF0DA	CRC calculation result register	CRCRESL	CRCRES	R/W	8/16	0xFF
0xF0DB		CRCRESH		R/W	8	0xFF
0xF0DC	Automatic CRC mode register	CRCMOD	-	R/W	8	0x00
0xF0DD	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

19.2.2 Automatic CRC Calculation Start Address Setting Register (CRCSAD)

Address: 0xF0D0
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CRCSAD															
Byte symbol	CRCSADH								CRCSADL							
Bit symbol	CRCSAD15	CRCSAD14	CRCSAD13	CRCSAD12	CRCSAD11	CRCSAD10	CRCSAD9	CRCSAD8	CRCSAD7	CRCSAD6	CRCSAD5	CRCSAD4	CRCSAD3	CRCSAD2	-	-
Access type	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	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCSAD is a special function register (SFR) used to set the start address of automatic CRC calculation.
CRCSAD is used as an address for the program code area and incremented during the automatic CRC calculation mode.

[Note]
- CRCSAD must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is invalid when the CRCAEN bit is "1".
- Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bits 1 and bit0 are ignored, fixed to “0” internally during the calculation.

19.2.3 Automatic CRC Calculation End Address Setting Register (CRCEAD)

Address: 0xF0D2
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CRCEAD															
Byte symbol	CRCEADH								CRCEADL							
Bit symbol	CRCEAD15	CRCEAD14	CRCEAD13	CRCEAD12	CRCEAD11	CRCEAD10	CRCEAD9	CRCEAD8	CRCEAD7	CRCEAD6	CRCEAD5	CRCEAD4	CRCEAD3	CRCEAD2	-	-
Access type	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	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

CRCEAD is a special function register (SFR) used to set the end address of automatic CRC calculation.

[Note]
- CRCEAD must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is invalid when the CRCAEN bit is "1".
- Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bits 1 and bit0 are ignored, fixed to “0” internally during the calculation.

19.2.4 Automatic CRC Calculation Start Segment Setting Register (CRCSSEG)

Address: 0xF0D4
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CRCSSEG							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	CRCSSEG3	CRCSSEG2	CRCSSEG1	CRCSSEG0
Access type	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

CRCSSEG is a special function register (SFR) used to set the start segment of automatic CRC calculation.
CRCSSEG is used as an address for the program code area and incremented during the automatic CRC calculation mode.

[Note]
CRCSSEG must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

19.2.5 Automatic CRC Calculation End Segment Setting Register (CRCESEG)

Address: 0xF0D6
Access: R/W
Access size: 8 bits
Initial value: 0xFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CRCESEG							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	CRCESEG3	CRCESEG2	CRCESEG1	CRCESEG0
Access type	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	1	1	1	1

CRCESEG is a special function register (SFR) used to set the end segment of automatic CRC calculation.

[Note]
CRCESEG must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

19.2.6 CRC Data Register (CRCDATA)

Address: 0xF0D8
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CRCDATA							
Bit symbol	-	-	-	-	-	-	-	-	CRCDATA7	CRCDATA6	CRCDATA5	CRCDATA4	CRCDATA3	CRCDATA2	CRCDATA1	CRCDATA0
Access type	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

CRCDATA is a special function register (SFR) used to set the CRC calculation data. Set it by eight bits.
One clock after writing data to the CRCDATA, the calculation result is stored in the CRC Calculation Result Register(CRCRES).

[Note]
CRCDATA must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

19.2.7 CRC Calculation Result Register (CRCRES)

Address: 0xF0DA
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CRCRES															
Byte symbol	CRCRESH								CRCRESL							
Bit symbol	CRCRES15	CRCRES14	CRCRES13	CRCRES12	CRCRES11	CRCRES10	CRCRES9	CRCRES8	CRCRES7	CRCRES6	CRCRES5	CRCRES4	CRCRES3	CRCRES2	CRCRES1	CRCRES0
Access type	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

CRCRES is a special function register (SFR). The CRC calculation result is stored by the hardware.
Set data to the CRCRES as an initial data for the CRC calculation.

[Note]
CRCRES must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

19.2.8 Automatic CRC Mode Register (CRCMOD)

Address: 0xF0DC
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CRCMOD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CRCAEN
Access type	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

CRCMOD is a special function register (SFR) used to control the automatic CRC calculation mode.

Description of bits

- **CRCAEN** (bit 0)
  The CRCAEN bit is used to enable the automatic CRC calculation mode.
  If the LSI switches to the HALT mode when CRCAEN is "1", CRC calculation is started for the program code area in the range set between the CRCSSEG and CRCESEG register and CRCSAD and CRCEAD register. When CRC calculation is completed, CRCAEN is set to "0."

CRCAEN	Description
0	Disables the automatic CRC calculation mode (initial value)
1	Enables the automatic CRC calculation mode

19.3 Description of Operation

19.3.1 CRC Calculation Mode

Set the initial value of CRC calculation in CRCRES. If 8-bit data is written to CRCDATA, the calculation result is stored in CRCRES on the next clock rising-edge. The CRC calculation result can be checked anytime by reading CRCRES. Figure 19-2 shows the operation timing of CRC calculation.

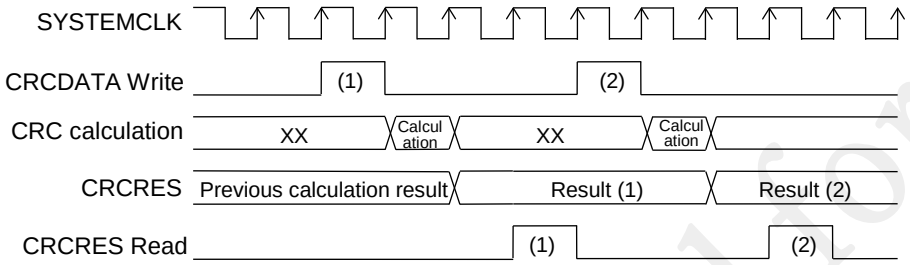


Figure 19-2 Timing of CRC Calculation.

[Note]
To perform CRC calculation when automatic CRC calculation is not completed, save the CRCRES value before calculation. After CRC calculation, move the saved value back to CRCRES and set CRCAEN to "1". When the LSI switches to the HALT mode, automatic CRC calculation can be restarted. The end addresses of the program code area are stored in CRCSAD and CRCSSEG. If the value is overwritten when CRCAEN is "0", data cannot be checked correctly.

19.3.2 Example of Use of CRC Calculation Mode

The following example shows serial transmission/reception (8 bits) using CRC calculation.
Figure 19-3 shows a flow for transmiting serial data with a CRC calculation result data.
The calculation result from 11-byte data starting with 0x21 is generated in the example.
Calculation setting data: 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F

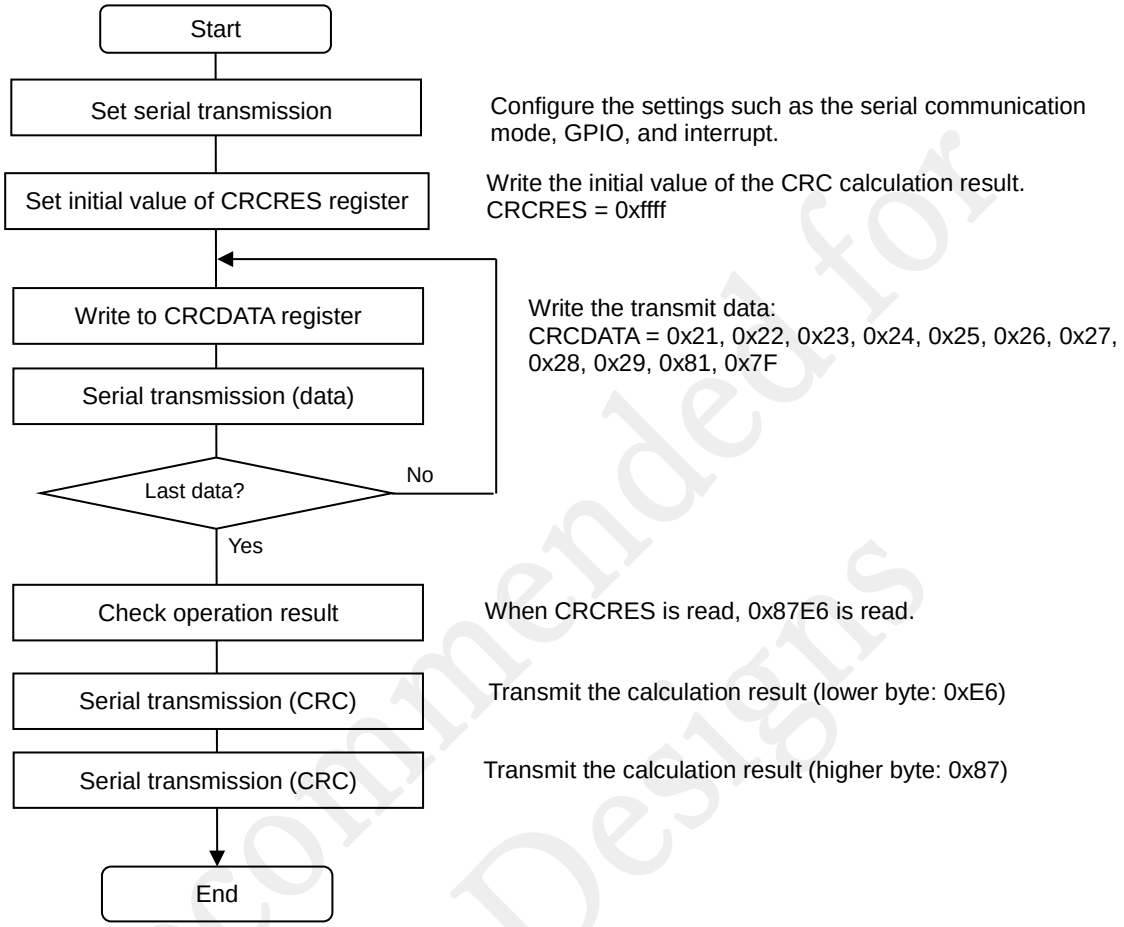


Figure 19-3 CRC Calculation Flow Example 1 (Serial Transmission)

Figure 19-4 shows a flow for receiving serial data with a CRC calculation result data.
It checks the calculation result from the 13 bytes in which the calculation result in the transmit side is added in the last two bytes.

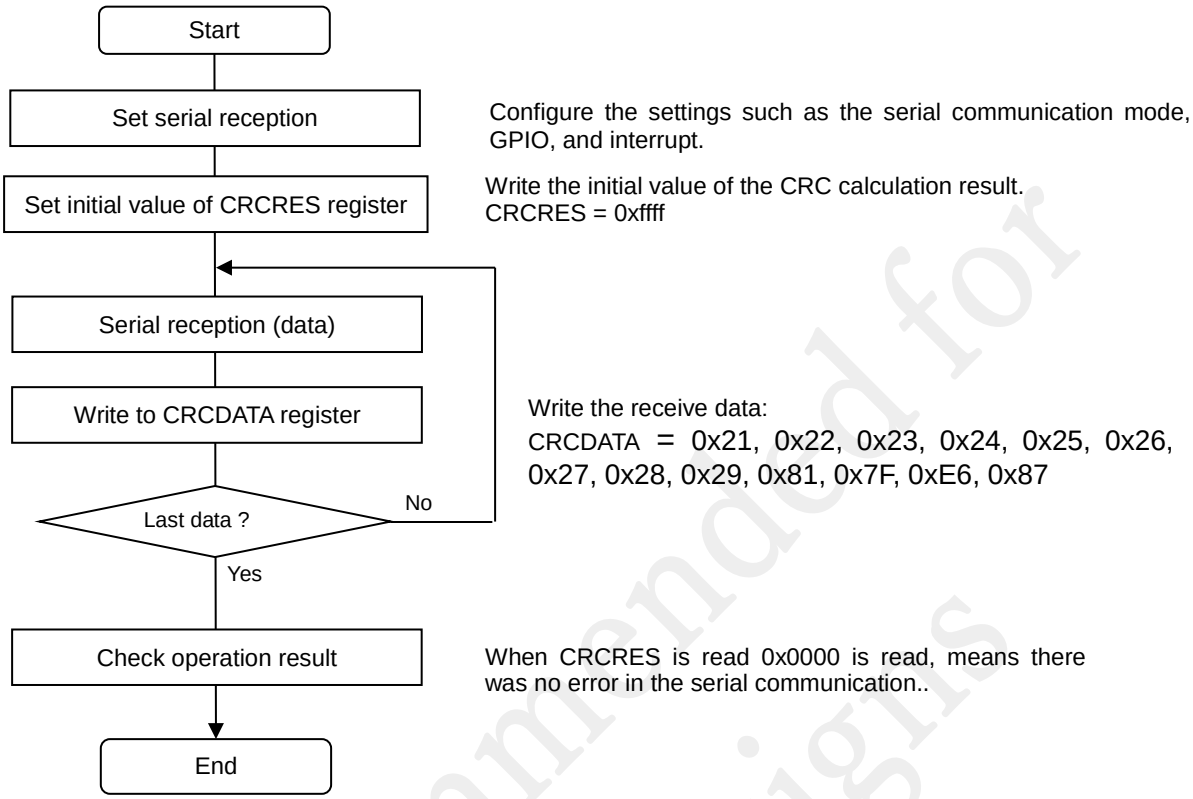


Figure 19-4 CRC Calculation Flow Example 2 (Serial Reception)

19.3.3 Automatic CRC Calculation Mode

Figure 19-5 shows an example of the specifications of automatic CRC calculation.
After the CRCAEN bit of CRCMOD is set to "1" and the CPU enterred in HALT/HALT-H mode, the hardware starts CRC calculation with data in the program code area from the start address specified in CRCSSEG and CRCSAD register to the end address specified in CRCESEG and CRCEAD register. If the HALT/HALT-H mode is released during calculation, calculation is halted. When the CPU gets back to HALT/HALT-H mode, the CRC calculation restarts from the last address before halted.
The data in CRCSSEG and CRCAD are automatically incremented every time the data is read from a address in the program code area. When the address data of CRCSSEG and CRCSAD and the address data of CRCESEG and CRCEAD coincide, the automatic CRC calculation is completed and the CRCAEN is set to "0".
The HALT/HALT-H mode is not released at the completion of automatic CRC calculation. Therefore, check the CRCAEN bit of CRCMOD after the HALT mode is released.

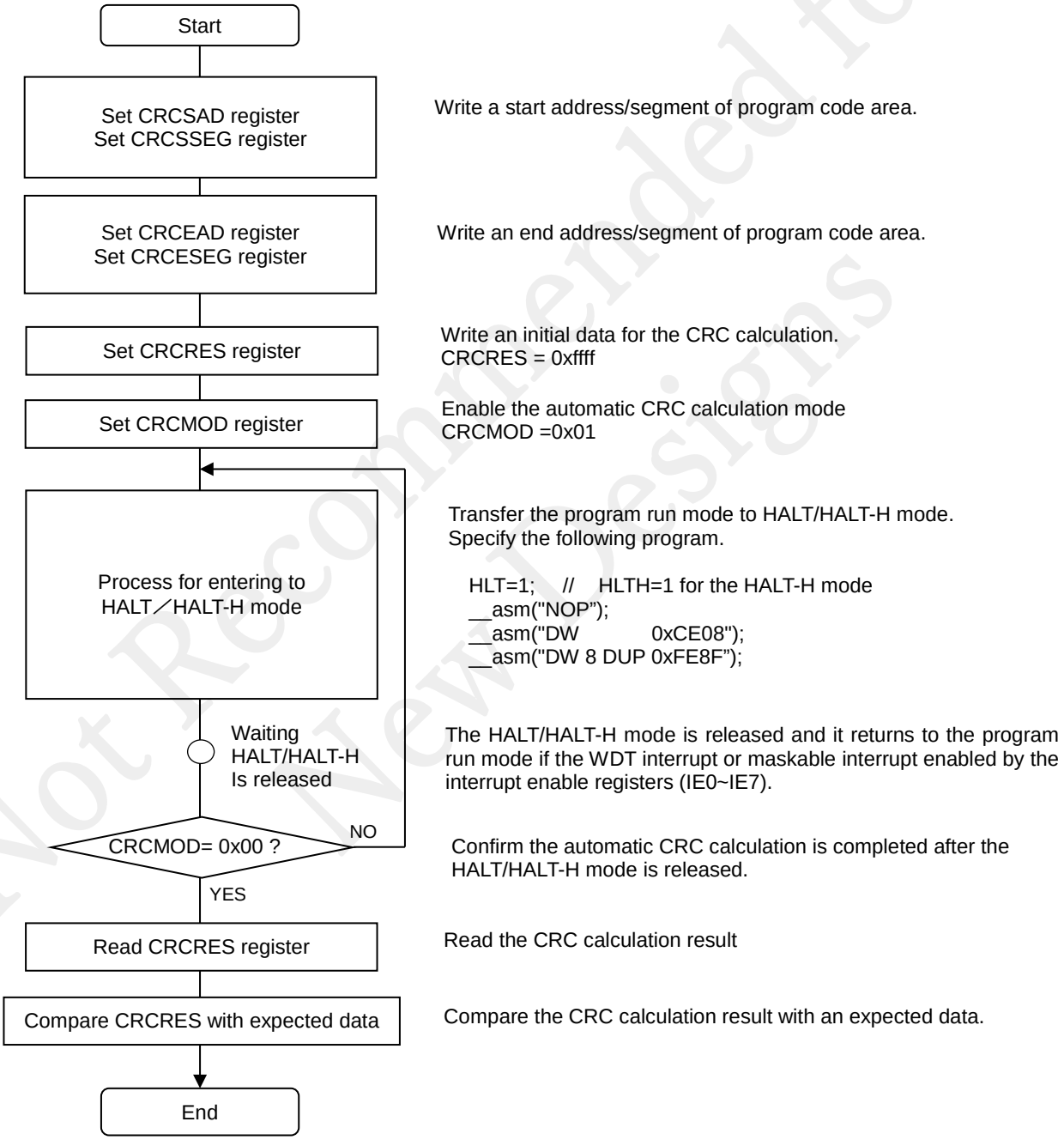


Figure 19-5 Example of Use of Automatic CRC Calculation

[Note]
When transferring to the HALT/HALT-H mode in the automatic CRC calculation mode, insert following program codes after setting the HLT/HALTH bit, otherwise the operation can not be guaranteed.

```
HLT=1;    // or HLTH=1;
__asm("NOP”);
__asm("DW 0CE08H");
__asm("DW 8 DUP 0FE8FH”);
```

Chapter 20 Analog Comparator

20. Analog Comparator

20.1 General Description

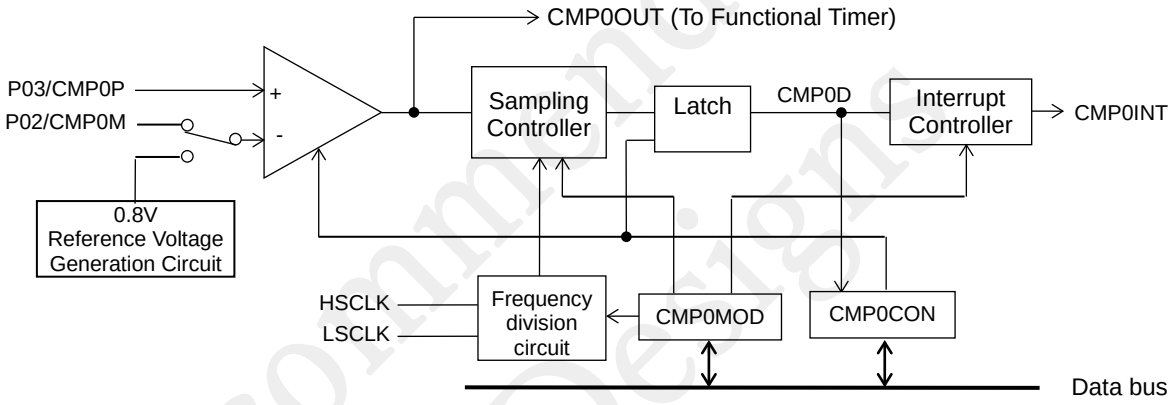
ML62Q1000 series has the analog comparator that compares the voltages of the two pins (CMP0P and CMP0M) input to the LSI.

20.1.1 Features

- Interrupt can be output based on the comparison results of the analog comparator
  (Allows selection of the interrupt disabled mode, falling-edge interrupt mode, rising-edge interrupt mode, or both-edge interrupt mode)
- Internal 0.8V reference voltage is available for the comparison
- The sampling is optional for the comparison result.
- The sampling frequency is selectable from LSCLK or HSCLK with dividing ratio 1/1 to 1/64
- Last comparison result (CMP0D) is retained when the analog comparator is stopped

20.1.2 Configuration

Figure 20-1 shows the configuration of the analog comparator.



CMP0CON : Comparator 0 control register
CMP0MOD : Comparator 0 mode register

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 secondary to octic functions of the GPIO.
For details about pin assignment and how to set the secondary to octic functions of the GPIO, see Chapter 17 "GPIO".

Pin name	I/O	Function
CMP0M	I	Analog comparator 0 non-inverting input
CMP0P	I	Analog comparator 0 inverting input

[Note]
When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to “0” as “Disable the input” and “Disable the output”, othewise a shoot-through current may flow.

20.2 Description of Registers

20.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF840	Comparator 0 control register	CMP0CON	-	R/W	8	0x00
0xF841	Reserved register	-	-	R	8	0x00
0xF842	Comparator 0 mode register	CMP0MODL	CMP0MOD	R/W	8/16	0x00
0xF843		CMP0MODH		R/W	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

20.2.2 Comparator 0 Control Register (CMP0CON)

Address: 0xF840
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								CMP0CON							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMP0D	CMP0EN
Access type	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

CMP0CON is a special function register (SFR) used to control the analog comparator.

Description of bits

- **CMP0EN** (bit 0)
  The CMP0EN bit is used to control enable or disable the operation of the comparator.

CMP0EN	Description
0	Disables operating the comparator (initial value)
1	Enables operating the comparator

- **CMP0D** (bit 1)
  The CMP0D bit is used to indicate the comparison result of the comparator output.
  Even when the comparator is stopped (CMP0EN is set to "0"), the last status is retained.

CMP0D	Description
0	CMP0P < CMP0M (initial value)
1	CMP0P > CMP0M

[Note]
An infuluence of the noise is reduceable by preventing the switching of neighboring pins while reading the COMP0D bit when the comparator enables.

20.2.3 Comparator 0 Mode Register (CMP0MOD)

Address: 0xF842
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CMP0MOD															
Byte symbol	CMP0MODH								CMP0MODL							
Bit symbol	-	-	-	-	-	-	-	CMP0VREF	-	CMP0DIV2	CMP0DIV1	CMP0DIV0	CMP0CS1	CMP0CS0	CMP0E1	CMP0E0
Access type	R	R	R	R	R	R	R	R/W	R	RW	RW	RW	RW	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP0MOD is a special function register (SFR) used to control the analog comparator.

Description of bits

- **CMP0E1 to CMP0E0** (bits 1-0)
  The CMP0E1 to CMP0E0 bits are used to select interrupt disabled mode, falling-edge interrupt mode, rising-edge interrupt mode, or both-edge interrupt mode.

CMP0E1	CMP0E0	Description
0	0	Interrupt disabled (initial value)
0	1	Falling-edge interrupt
1	0	Rising-edge interrupt
1	1	Both-edge interrupt

- **CMP0CS1 to CMP0CS0** (bits 3-2)
  The CMP0CS1 to CMP0CS0 bits are used to select with/without sampling for the comparator comparison result and select the sampling clock.

CMP0CS1	CMP0CS0	Description
0	0	Detects without sampling (initial value)
0	1	Sampling with HSCLK
1	0	Sampling with LSCLK
1	1	Do not use (Detects without sampling)

- **CMP0DIV2 to CMP0DIV0** (bits 6-4)
  The CMP0DIV2 to CMP0DIV0 bits are used to select the sampling clock dividing ratio for the comparator comparison result.

CMP0DIV2	CMP0DIV1	CMP0DIV0	Description
0	0	0	1/1 (initial value)
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	1/64
1	1	1	Do not use (1/1)

- **CMP0VREF** (bit 8)
  The CMP0VREF bit is used to select the reference voltage of the comparator.

CMPVREF	Description
0	Uses a reference voltage input from the CMP0M pin (initial value)
1	Uses the internal 0.8V reference voltage

[Note]
- In the STOP/STOP-D/HALT-H mode, no sampling is performed regardless of the value set in CMP0CS1 and CMP0CS0 since the sampling clock stops.
  In this switching from "with sampling" to "without sampling", there is a time period (*) in which interrupts gets disabled.
  When entering to STOP/STOP-D/HALT-H mode, specify the external interrupt as "without sampling".
  After returning from STOP/STOP-D/HALT-H mode, specify the CMP0CS1 and CMP0CS0 as "with sampling" if needed.
  (*) When entering to STOP/STOP-D mode: Max.30us
  When returing from STOP/STOP-D mode: Max.250us(When LSCLK is selected for CPU),
  Max.2.5ms(When HSCLK is selected for CPU)
  When returing from HALT-H mode: Max.2.5ms(When HSCLK is selected for CPU)
- When the HSCLK is selected and ENOSC bit of FCON register is "0", the sampling function is not available.
- CMP0MOD must be written when the comparator operation is disabled (CMP0EN = 0). If it is written when the comparator operation is enabled, the comparison result is not guaranteed.

20.3 Description of Operation

20.3.1 Operation of Analog Comparator

The analog comparator compares “the input voltages of the CMP0P pin to CMP0M pin” or “the input voltages of the CMP0P pin to the internal 0.8V reference voltage” to get the comparison result throuh a sampling clock.
The comparsion result is readable from CMP0D bit of the comparator 0 control register (CMP0CON). When the CMP0D bit is "1" it indicates that the input voltage of CMP0P pin is higher than the input voltage of CMP0M pin(or higher than the internal 0.8V). When CMP0D is "0" it indicates that the input voltage of CMP0P pin is lower than the input voltage of the CMP0M pin(or lower than the internal 0.8V).
When CMP0EN bit is set to "1", the analog comparator is enabled. When the CMP0EN bit is set to "0", the analog comparator is disabled.
Read the CMP0D bit 100 μs or longer after "1" is written to the CMP0EN bit, because it takes 100 μs for the analog comparator to stabilize.
Figure 20-2 shows an example of the operation time chart.

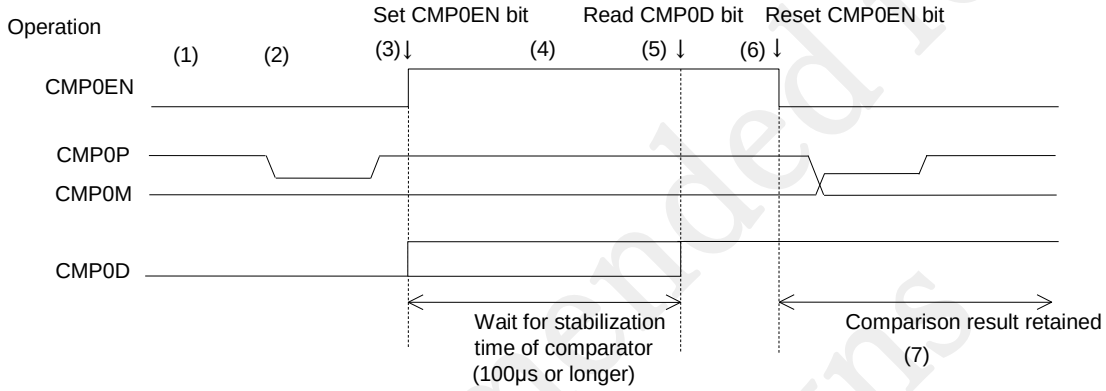


Figure 20-2 Operation Time Chart

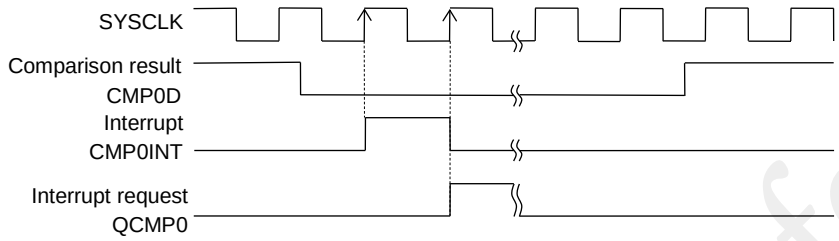
The operations in Figure 20-2 are as follows.

(1) Configure the following settings before operating the comparator.
- Set the GPIO as high-impedance output.
- When using the high-speed clock for the sampling clock, write "1" to the ENOSC bit of the clock control register (FCON).
- When using interrupt, write "1" to the ECMP0 bit of the interrupt enable register 45 (IE45).
- Use the block control register (BCKCON) to supply the clock.
- Use the block reset control register (BRECON) to release the comparator reset.

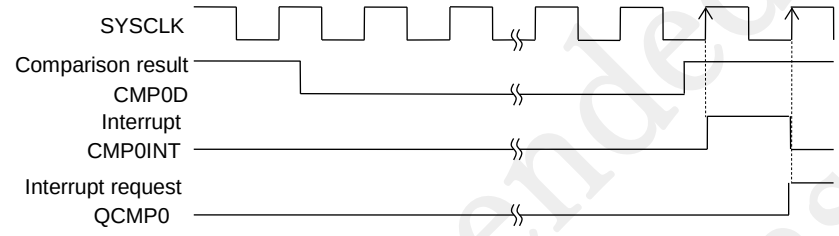
(2) Select the interrupt mode and sampling by CMP0MOD.
(3) Write "1" to CMP0EN to enable the comparator operation.
(4) Wait the settling time (100 μs or longer) of the comparator.
(5) Read the comparison result (CMP0D).
(6) Write "0" to CMP0EN to disable the comparator operation. At the same time, the result is retained.
(7) CMP0D can be read after "0" is written to CMP0EN because CMP0D holds the comparison result at the time when "0" is written to CMP0EN.

20.3.2 Interrupt Request

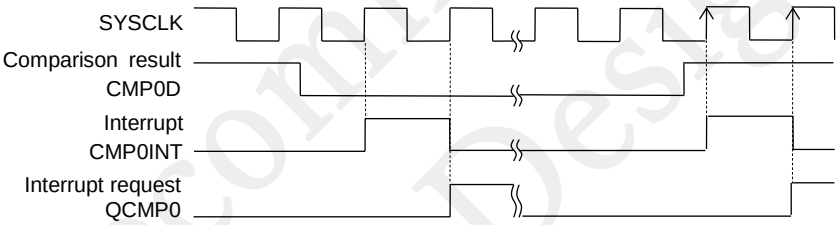
When an interrupt edge selected by the comparator 0 mode register (CMP0MOD) is detected on the comparison result of the comparator, a comparator interrupt (CMP0INT) is generated.
Figure 20-3 shows the interrupt generation timing in rising-edge interrupt mode, in falling-edge interrupt mode, and in both-edge interrupt mode without sampling, and in rising-edge interrupt mode with sampling.
Figure 20-4 shows the interrupt generation timing in the STOP/STOP-D mode.



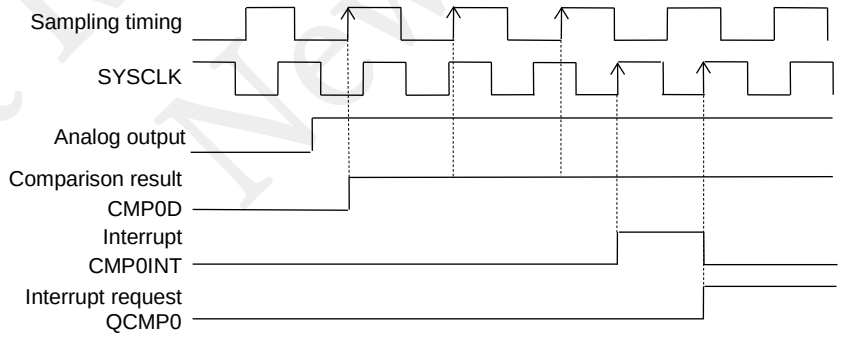
(a) When falling-edge interrupt mode without sampling is selected



(b) When rising-edge interrupt mode without sampling is selected

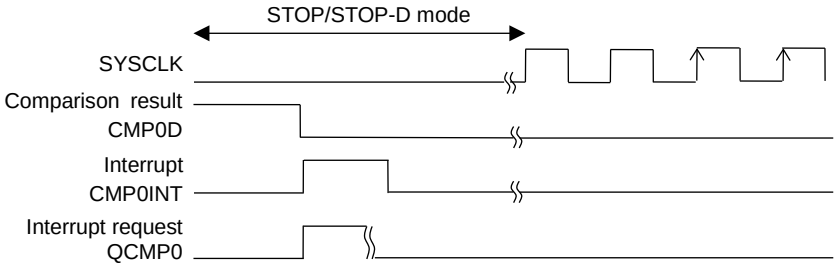


(c) When both-edge interrupt mode without sampling is selected



(d) When rising-edge interrupt mode with sampling is selected

Figure 20-3 Analog Comparator Interrupt Generation Timing



(a) When falling-edge interrupt mode is selected

Figure 20-4 Analog Comparator Interrupt Generation Timing in STOP/STOP-D Mode

Chapter 21 D/A Converter

21. D/A Converter

21.1 General Description

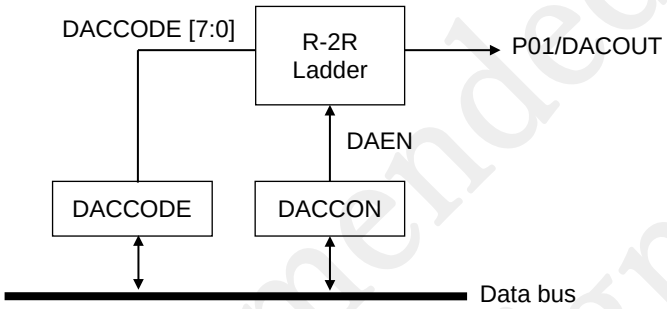
ML62Q1000 series has one channel 8-bit resolution D/A converter that converts digital input signals to analog signals. See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

21.1.1 Features

- 8-bit resolution x 1 ch
- R-2R ladder method
- Analog output voltage
  - 8-bit resolution: V_{DD} x (SFR setting value) / 256
  - Output impedance: Approx. 6kΩ

21.1.2 Configuration

Figure 21-1 shows the configuration of the D/A converter.



DACEN : D/A converter enable register
DACCODE : D/A converter code register

Figure 21-1 Configuration of D/A Converter Circuit

21.1.3 List of Pins

The output pin of the D/A converter is assigned to the secondary to octic function of the GPIO.
For details about the pin assignment and how to set the secondary to octic functions of the GPIO, see Chapter 17 "GPIO".

Pin name	I/O	Function
DACOUT	O	D/A converter output pin

[Note]
When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to “0” as “Disable the input” and “Disable the output”, othewise a shoot-through current may flow.

21.2 Description of Registers

21.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF860	D/A converter enable register	DACEN	-	R/W	8	0x00
0xF861	Reserved register	-	-	R	8	0x00
0xF862	D/A converter code register	DACCODE	-	R/W	8	0x00
0xF863	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

21.2.2 D/A Converter Enable Register (DACEN)

Address: 0xF860
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								DACEN							
Bit symbol	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'	DAEN
Access type	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

DACEN is a special function register (SFR) used to select the control of the D/A converter.

Description of bits

- **DAEN** (bit 0)
  The DAEN bit is used to enable or disable the operation of the D/A converter.

DAEN	Description
0	Disables operating the D/A converter (initial value)
1	Enables operating the D/A converter

[Note]
- When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to “0” as “Disable the input” and “Disable the output”, othewise a shoot-through current may flow.
- An infuluence of the noise is reduceable by preventing the switching of neighboring pins while the D/A conversion is enabled.

21.2.3 D/A Converter Code Register (DACCODE)

Address: 0xF862
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								DACCODE							
Bit symbol	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
Access type	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

DACCODE is a special function register (SFR) used to set the output code of the D/A converter circuit.

Description of bits

- **d7 to d0** (bits 7-0)
  The d7 to d0 bits are used to set the conversion value of the D/A converter.
  A voltage of V_{DD} x DACCODE / 256 is output.

d7 to d0	Description
0x00	Outputs V_{SS} (initial value)
0x01 – 0xFF	Outputs a voltage = V_{DD} x (decimal code of DACCODE register / 256)

21.3 Description of Operation

21.3.1 Operation of D/A Converter

Figure 21-2 shows the operation flow of D/A convertor and Figure 21-3 shows the timing diagram of D/A conversion operation.

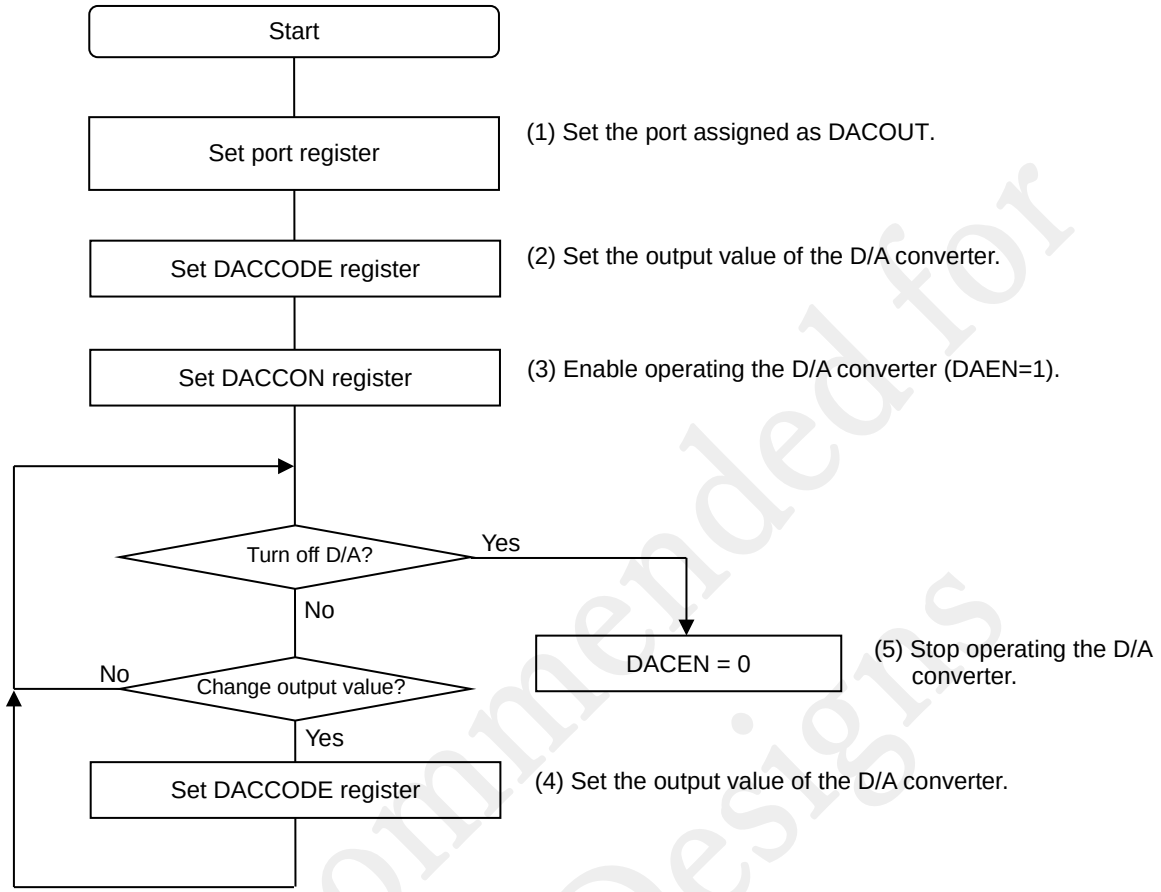
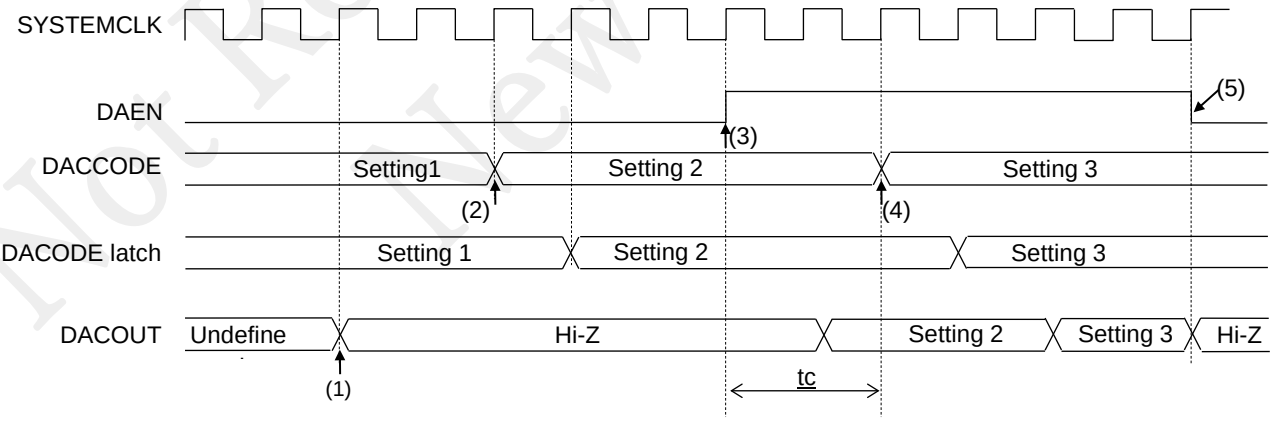


Figure 21-2 D/A Convertor Operation Flow



tc : D/A converter conversion cycle

Figure 21-3 Timing Diagram of D/A Conversion Operation

Chapter 22 Voltage Level Supervisor (VLS)

22. Voltage Level Supervisor (VLS)

22.1 General Description

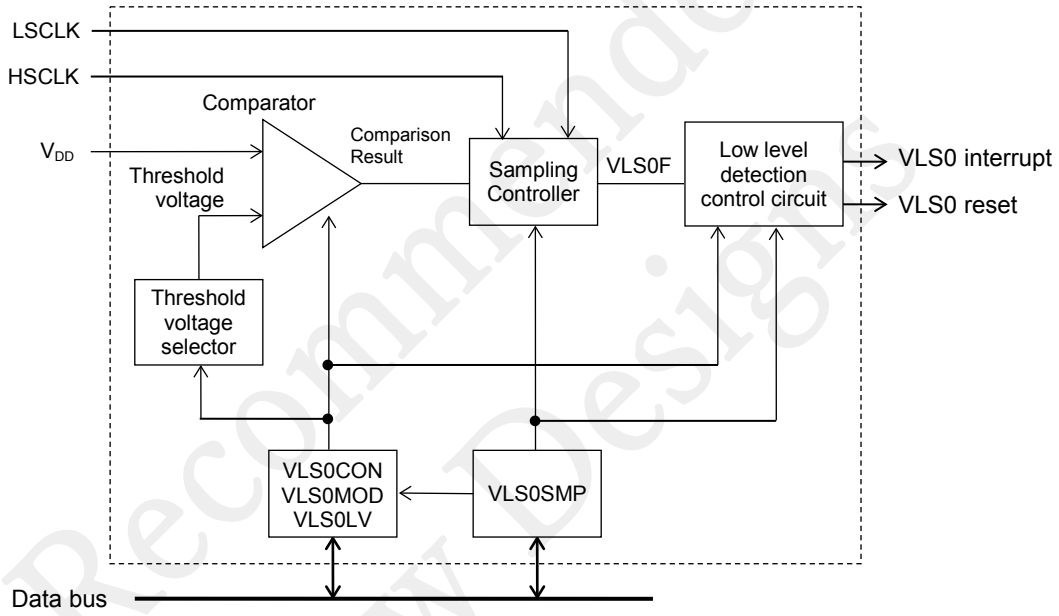
ML62Q1000 series has the Voltage Level Supervisor (VLS) that detects whether the voltage level of VDD is lower than the specified threshold voltage.

22.1.1 Features

- Accuracy: ±4 %
- Threshold voltage: Selectable from 12 values (1.85 to 4.00 V)
- Supervisor mode (continuous operation) or single mode (one operation) can be selected for the operation mode
- Can be used as the voltage level detection reset (VLS0 reset)
- Can be used as the voltage level detection interrupt (VLS0 interrupt)
- Can be initialized only by the power-on reset (POR) or pin reset

22.1.2 Configuration

The voltage level supervisor (VLS) consists of a comparator, a sampling control circuit, and a low level detection control circuit. Figure 22-1 shows the configuration of the VLS.



VLS0CON : Voltage level supervisor 0 control register
VLS0MOD : Voltage level supervisor 0 mode register
VLS0LV : Voltage level supervisor 0 level register
VLS0SMP : Voltage level supervisor 0 sampling register

Figure 22-1 Configuration of Voltage Level Supervisor

22.2 Description of Registers

22.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF850	Voltage level supervisor 0 control register	VLS0CON	-	R/W	8	0x00
0x851	Reserved register	-	-	R	8	0x00
0x852	Voltage level supervisor 0 mode register	VLS0MOD	-	R/W	8	0x00
0x853	Reserved register	-	-	R	8	0x00
0x854	Voltage level supervisor 0 level register	VLS0LV	-	R/W	8	0x00
0x855	Reserved register	-	-	R	8	0x00
0x856	Voltage level supervisor 0 sampling register	VLS0SMP	-	R/W	8	0x00
0x857	Reserved register	-	-	R	8	0x00

22.2.2 Voltage Level Supervisor Control Register (VLS0CON)

Address: 0xF850
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								VLS0CON							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	VLS0RF	VLS0F	VLS0EN
Access type	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

VLS0CON is a special function register (SFR) used to control the voltage level supervisor.
This register cannot be reset by anything other than the Power On Reset(POR) and RESETN pin reset.

Description of bits

- **VLS0EN** (bit 0)
  The VLS0EN bit is used to control the VLS0 circuit operation.
  When VLS0EN is set to "1", the VLS0 circuit operation is enabled and when it is set to "0", the VLS0 circuit operation is disabled.
  In the single mode, it is automatically set to "0" and the VLS0 circuit operation is disabled at the completion of judgment.

VLS0EN	Description
0	Enables operating the VLS0 circuit (initial value)
1	Disables operating the VLS0 circuit

- **VLS0F** (bit 1)
  VLS0F is the voltage level detection flag.
  It is "0" when the power supply voltage (V_{DD}) is higher than the threshold voltage, or "1" when the power supply voltage is lower than the threshold voltage.
  VLS0F retains the last threshold result.
  When "1" is written to the VLS0F bit, the VLS0F bit is cleared to "0".
  VLS0F is cleared to "0" at the start of VLS0.

VLS0F	Description
0	Higher than the threshold voltage (initial value)
1	Lower than the threshold voltage

- **VLS0RF** (bit 2)
  The VLS0RF flag is used to indicate whether the voltage level detection result is valid or not.
  When the threshold voltage value becomes valid (readable from CPU), this becomes "1".
  VLS0RF is valid only when the VLS0 running mode is the supervisor mode.
  VLS0RF is fixed to "0" when the VLS0 running mode is the single mode.

VLS0RF	Description
0	The VLS0 circuit is stopped or VLS0 is being stabilized (initial value)
1	The VLS0 detection result is valid

[Note]
- During the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode. When the VLS0RF bit is set to "0", the LSI cannot enter the STOP/STOP-D mode.
- During the single mode, the LSI cannot enter the STOP/STOP-D mode. When the single mode operation is completed (make sure that the VLS0EN bit is set to "0"), enable the STOP/STOP-D mode.
- The VLS0 interrupt is not available to wake up from the STOP/STOP-D mode. Enter the STOP/STOP-D mode after disabling the EVLS0 bit of interrupt enable register IE0.
- Even if a reset (VLS0 reset, WDT reset, ROM unused area access reset, or RAM parity error reset) other than the POR and pin reset is output, VLS0 remains operating.

22.2.3 Voltage Level Supervisor Mode Register (VLS0MOD)

Address: 0xF852
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								VLS0MOD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	VLS0AMD1	VLS0AMD0	-	-	VLS0SEL1	VLS0SEL0
Access type	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

VLS0MOD is a special function register (SFR) used to control the operation mode of the voltage level detection function.
Set this register only when the VLS0 circuit is stopped (VLS0EN = "0").
This register cannot be reset by anything other than the Power On Reset(POR) and RESETN pin reset.

Description of bits

- **VLS0SEL1 to VLS0SEL0** (bits 1-0)
  The VLS0SEL1 to VLS0SEL0 bits are used to control enable/disable of the VLS0 reset/VLS0 interrupt request functions when the voltage level of VDD is lower than the threshold voltage.
  The reset function is enabled only in the supervisor mode. A reset does not occur when the reset function is selected in the single mode 1 and sigle mode 2.

VLS0SEL1	VLS0SEL0	Description
0	0	Reset function: disable, Interrupt request function: disable (initial value)
0	1	Reset function: enable, Interrupt request function: disable
1	0	Reset function: disable, Interrupt request function: enable
1	1	Do not use (Reset function: enable, Interrupt request function: disable)

[Note]
Select always “With sampling” in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.

- **VLS0AMD1 to VLS0AMD0** (bits 5-4)
  The VLS0AMD1 to VLS0AMD0 bits are used to set the VLS0 running mode.
  In the single mode 1 or sigle mode 2, the voltage level of VDD is monitored only once while in the supervisor mode the voltage level of VDD is always monitored.
  The monitoring result can be checked by reading the VLS0F bit of the voltage level supervisor control register (VLS0CON).

VLS0AMD1	VLS0AMD0	Description
0	0	Single mode 1 (initial value) The single mode 1 monitors the voltage level of VDD only once. When VLS0SEL[1:0] is “0x02” the interrupt occurs after every monitoring the VDD voltage. The result can be checked by reading VLS0F bit of VLS0CON register. No interrupt occurs on the other conditions of VLS0SEL[1:0].
0	1	Single mode 2 The single mode 2 monitors the voltage level of VDD only once. When VLS0SEL[1:0] is “0x02” the interrupt occurs if the VDD voltage is lower than the threshold voltage specified in VSL0LV[3:0]. The result can be checked by reading VLS0F bit of VLS0CON register. No interrupt occurs on the other conditions of VLS0SEL[1:0].
1	*	Supervisor mode The supervisor mode always monitors the voltage level of VDD. The interrupt or reset occurs depending on the conditions of VSL0SEL[1:0].

*0 or 1(don’t care)

[Note]
- During the single mode 1 and single mode 2, the LSI cannot enter the STOP/STOP-D mode. Make sure that the VLS0EN bit is set to "0" before enabling the STOP/STOP-D mode.
- During the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode. When the VLS0RF bit is set to "0", the LSI cannot enter the STOP/STOP-D mode.
- The VLS0 interrupt is not available to wake up from the STOP/STOP-D mode. Enter the STOP/STOP-D mode after disabling the EVLS0 bit of interrupt enable register IE0.
- Select always “With sampling” in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.
- In the the supervisor mode, the interrupt occurs even when the voltage level of V_{DD} changes from lower to higher than the threshold voltage.

22.2.4 Voltage Level Supervisor 0 Level Register (VLS0LV)

Address: 0xF854
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								VLS0LV							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	VLS0LV3	VLS0LV2	VLS0LV1	VLS0LV0
Access type	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

VLS0LV is a special function register (SFR) used to set the detection voltage.
Set this register only when the VLS0 circuit is stopped (VLS0EN = "0").
This register cannot be reset by anything other than the Power On Reset(POR) and RESETN pin reset.

Description of bits

- **VLS0LV3 to VLS0LV0** (bits 3-0)
  The VLS0LV3 to VLS0LV0 bits are used to select the threshold voltage (VVLSF) of VLS0 while the VDD is falling. VLS0 has hysteresis characteristics. For the characteristics of threshold voltage when the power is falling and rising, see the data sheet of each product.

VLS0LV3	VLS0LV2	VLS0LV1	VLS0LV0	Description
0	0	0	0	4.00 ±4 % (initial value)
0	0	0	1	3.70 ±4 %
0	0	1	0	3.05 ±4 %
0	0	1	1	2.95 ±4 %
0	1	0	0	2.85 ±4 %
0	1	0	1	2.75 ±4 %
0	1	1	0	2.65 ±4 %
0	1	1	1	2.55 ±4 %
1	0	0	0	2.45 ±4 %
1	0	0	1	2.05 ±4 %
1	0	1	0	1.95 ±4 %
1	0	1	1	1.85 ±4 %
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

22.2.5 Voltage Level Supervisor 0 Sampling Register (VLS0SMP)

Address: 0xF856
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								VLS0SMP							
Bit symbol	-	-	-	-	-	-	-	-	-	VLS0DIV2	VLS0DIV1	VLS0DIV0	VLS0SM1	VLS0SM0	-	-
Access type	R	R	R	R	R	R	R	R	R	R/W	R/W	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

VLS0SMP is a special function register (SFR) used to control the sampling function for the VLS0 analog output of the voltage level detection function.
Set this register only when the VLS0 circuit is stopped (VLS0EN = "0").
This register cannot be reset by anything other than the Power On Reset(POR) and RESETN pin reset.

Description of bits

- **VLS0SM1 to VLS0SM0** (bits 3-2)
  The VLS0CS1 to VLS0CS0 bits are used to select the sampling clock for the VLS0 analog output.

VLS0SM1	VLS0SM0	Description
0	0	Without sampling (initial value) Do not select when enabling the VLS0 interrupt function in the supervisor mode.
0	1	With sampling by HSCLK
1	0	With sampling by LSCLK
1	1	Without sampling Do not select when enabling the VLS0 interrupt function in the supervisor mode.

- **VLS0DIV2 to VLS0DIV0** (bits 6-4)
  The VLS0DIV2 to VLS0DIV0 bits are used to select the sampling clock dividing ratio for the VLS0 analog output.

VLS0DIV2	VLS0DIV1	VLS0DIV0	Description
0	0	0	1/1 (initial value)
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	1/64
1	1	1	1/1

22.3 Description of Operation

VLS0 can read whether V_{DD} is lower or higher than the threshold voltage set in the CPU from the SFR. It also outputs a VLS0 interrupt or VLS0 reset when V_{DD} is lower than the specified threshold voltage.
VLS0 has hysteresis characteristics. For the characteristics of threshold voltage when the power is falling and rising, see the data sheet of each product.

The following two operation modes are provided:

Supervisor mode:

When "1" is written to VLS0EN, the VLS0 operation is enabled and it starts monitoring the VDD voltage.
When the result becomes valid, it is notified by data in VLS0RF bit and the monitoring still continues.
VLS0 interrupt or VLS0 reset is generateable if the VDD voltage is lower than the threshold voltage during the constant monitoring.
Select always “With sampling” in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.

Single mode:

When "1" is written to VLS0EN, the VLS0 operation is enabled and it starts monitoring the VDD voltage.
When the result becomes valid, "0" is automatically written to VLS0EN to end the monitoring.
VLS0 interrupt is generateable on following conditions in each mode.
Single mode 1: The interrupt occurs every VSL0 operation regardless the result.
Single mode 2: The interrupt occurs if the VDD voltage is lower than the threshold voltage(selected in the VLS0LV) when the monitoring is completed.

22.3.1 Supervisor Mode

When setting VLS0EN with VLS0AMD1-0 set to "10B" or "11B", the supervisor mode is activated.
The supervisor mode starts controlling by checking the ready flag (VLS0RF) from the software. It is useful for using the low voltage detection interrupt/reset with always VLS0 enabled. The detection flag (VLS0F)/interrupt/reset is masked until the ready flag (VLS0RF) is asserted.

The ready flag (VLS0RF) is asserted when about 300 us (+ sampling clock x 3 when the sampling is enabled) passes after setting "1" to VLS0EN from the software. The software waits that "1" is set to VLS0RF and reads the value of the voltage level detection flag (VLS0F).
When VDD becomes lower than the specified threshold voltage, a VLS0 interrupt or VLS0 reset is output.
To turn off the VLS0 function, set VLS0EN to "0" from the software.

Figure 22-2 shows an operation timing diagram when the VLS0 reset with sampling is selected.

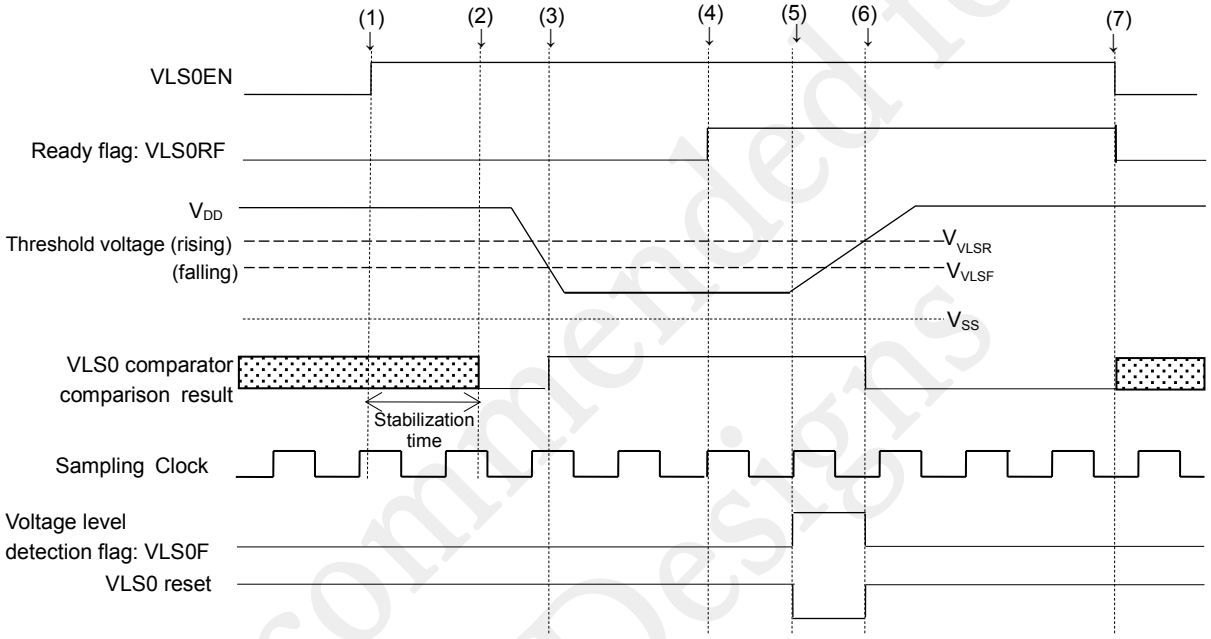


Figure 22-2 Operation Timing Diagram (when the VLS0 reset with sampling is selected)

(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) The VLS0 analog output is stabilized.
(3) V_{DD} becomes lower than the specified threshold voltage (V_{VLSF}).
(4) When the VLS0 analog output is stabilized, the ready flag (VLS0RF) is set to "1" after three sampling clocks.
(5) The voltage level detection flag (VLS0F) is set to "1" to output a VLS0 reset because VLS0 analog output becomes lower than the specified threshold voltage (V_{VLS0}) for as much time as three sampling clocks.
(6) VLS0 analog output is back to higher than the threshold voltage (V_{VLSR}). At the same time, the voltage level detection flag (VLS0F) is set to "0" to release the VLS0 reset.
(7) "0" is written to VLS0EN from the CPU to disable VLS0.

[Note]
- When VLS0 is being stabilized, the LSI cannot shift to the STOP/STOP-D mode. After starting the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode.
- The VLS0 interrupt is not available to wake up from the STOP/STOP-D mode. Enter the STOP/STOP-D mode after disabling the EVLS0 bit of interrupt enable register IE0.
- Select always “With sampling” in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.

Figure 22-3 shows an operation timing diagram when the VLS0 interrupt with sampling is selected.

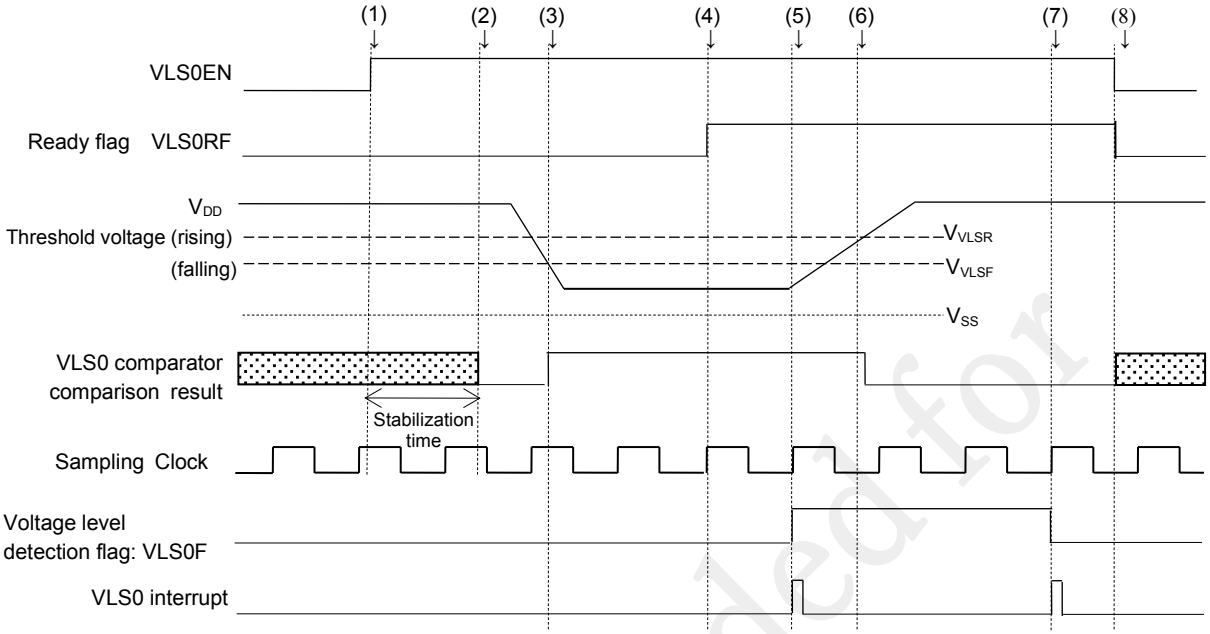


Figure 22-3 Operation Timing Diagram (when the VLS0 interrupt with sampling is selected)

(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) The VLS0 analog output is stabilized.
(3) V_{DD} becomes lower than the specified threshold voltage (V_{VLSF}).
(4) When the VLS0 analog output is stabilized, the ready flag (VLS0RF) is set to "1" after three sampling clocks.
(5) The voltage level detection flag (VLS0F) is set to "1" to output a VLS0 interrupt because VLS0 analog output becomes lower than the specified threshold voltage (V_{VLSR}) for as much time as three sampling clocks.
(6) VLS0 analog output is back to higher than the threshold voltage (V_{VLSR}).
(7) The voltage level detection flag (VLS0F) is set to "0" to output a VLS0 interrupt because VLS0 analog output becomes higher than the specified threshold voltage (V_{VLSR}) for as much time as three sampling clocks.
(8) "0" is written to VLS0EN from the CPU to disable VLS0.

[Note]
- When VLS0 is being stabilized, the LSI cannot enter the STOP/STOP-D mode. After starting the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode.
- The VLS0 interrupt is not available to wake up from the STOP/STOP-D mode. Enter the STOP/STOP-D mode after disabling the EVLS0 bit of interrupt enable register IE0.
- Select always “With sampling” in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.
- When VLS0 is stopped (VLS0EN = "0") during the low voltage detection state (VLS0F = "1"), a VLS0 interrupt occurs.

22.3.2 Single Mode

When writing "1" to VLS0EN with VLS0AMD[1:0] set to "00B" or "01B", the single mode is activated.
In the single mode, the software waits a VLS0 interrupt for judgment. It is useful for intermittently checking V_{DD}.
An interrupt in the single mode is different depending on the operation mode. An interrupt indicates the completion of judgement in the single mode 1 and the low voltage detection in the single mode 2.

Single mode 1
The result of monitoring VDD voltage is stored in the voltage level detection flag (VLS0F), about 300 us after (+ sampling clock x 3 when the sampling is enabled) setting "1" to VLS0EN by the software. Then, the VLS0 interrupt occurs and the VLS0EN is automatically reset.

Single mode 2
The result of monitoring VDD voltage is stored in the voltage level detection flag (VLS0F), about 300 us after (+ sampling clock x 3 when the sampling is enabled) setting "1" to VLS0EN by the software. Then, the VLS0 interrupt occurs if the VLS0F is “1” and the VLS0EN is automatically reset (When VLS0F is "0" the VLS0 interrupt does not occur).

Figures 22-4 shows an operation timing diagram when the single mode 1 without sampling is selected.

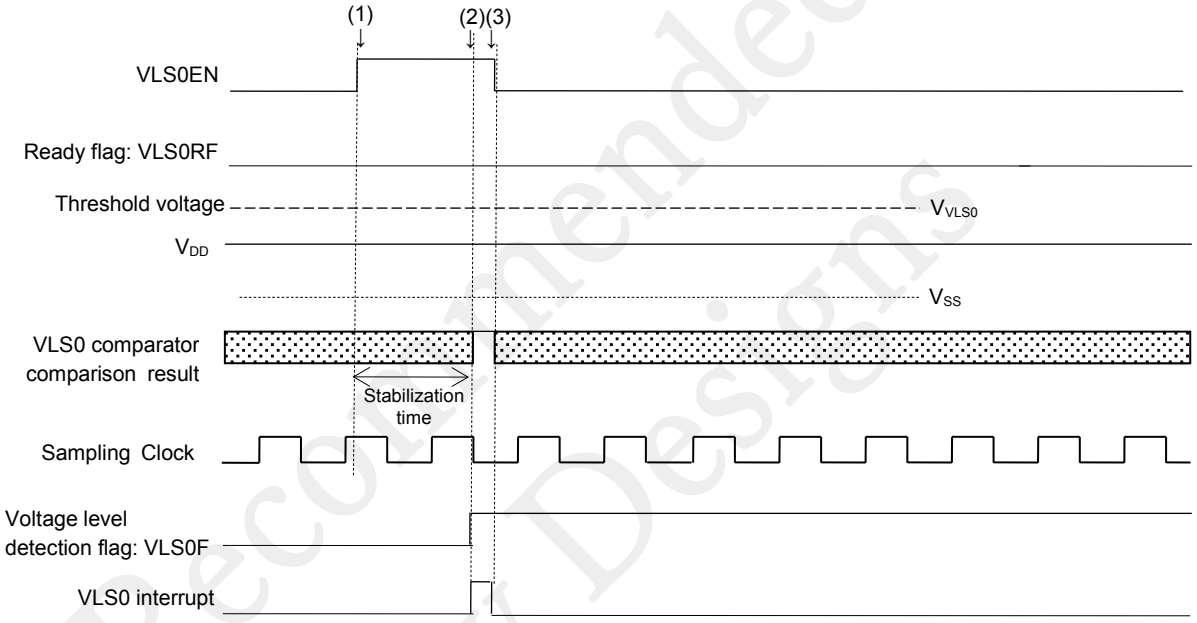
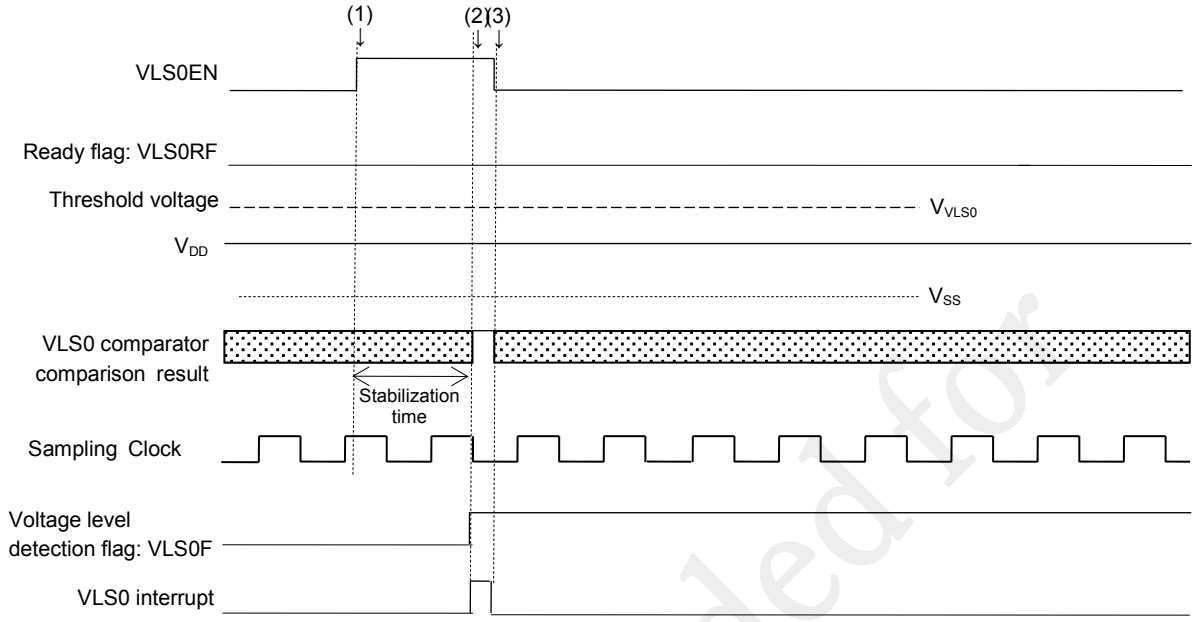


Figure 22-4 Operation Timing Diagram (when the single mode 1 without sampling is selected)

(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) When VLS0 analog output is stabilized, the voltage level detection flag (VLS0F) is set to "1" to output a VLS0 interrupt (completion of judgment) because V_{DD} is judged to be lower than the specified threshold voltage (V_{VLS0}).
A VLS0 interrupt (completion of judgment) is output regardless of the judgment result of V_{DD}.
(3) "0" is automatically written to VLS0EN to disable VLS0.
(4) The voltage level detection flag (VLS0F) is read from the CPU to check the judgment result.

[Note]
During the single mode, the LSI cannot enter the STOP/STOP-D mode. When the single mode operation is completed (make sure that the VLS0EN bit is set to "0"), enable the STOP/STOP-D mode.

Figures 22-5 shows an operation timing diagram when the single mode 2 without sampling is selected.



(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) When VLS0 analog output is stabilized, the voltage level detection flag (VLS0F) is set to "1" to output a VLS0 interrupt (low voltage detection) because V_{DD} is judged to be lower than the specified threshold voltage (V_{VLS0}).
If V_{DD} is judged to be higher than the specified threshold voltage (V_{VLS0}), the voltage level detection flag (VLS0F) is set to "0" and a VLS0 interrupt (low voltage detection) is not output.
(3) "0" is automatically written to VLS0EN to disable VLS0.

Figure 22-5 Operation Timing Diagram (when the single mode 2 without sampling is selected)

[Note]
- When VLS0 is being stabilized, the LSI cannot enter the STOP/STOP-D mode. After starting the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode.
- When VLS0 is stopped (VLS0EN = "0") during the low voltage detection state (VLS0F = "1"), a VLS0 interrupt occurs.

Figures 22-6 shows an operation timing diagram when the single mode 1 with sampling is selected.

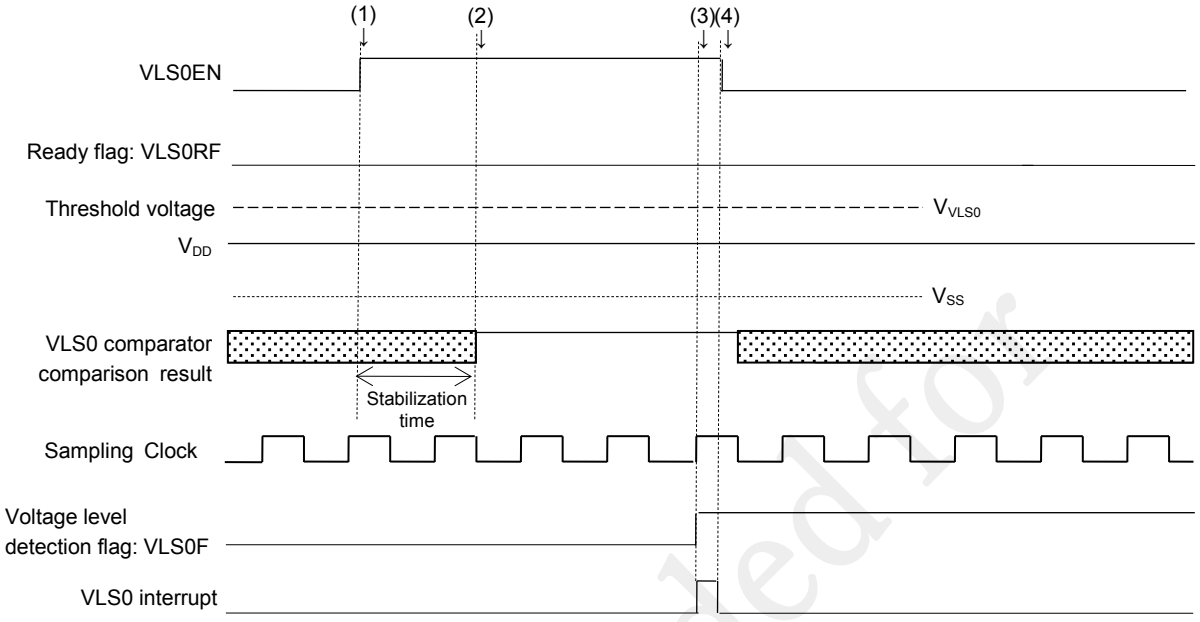


Figure 22-6 Operation Timing Diagram (when the single mode 1 with sampling is selected)

(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) The VLS0 analog output is stabilized.
(3) The voltage level detection flag (VLS0F) is set to "1" to output a VLS0 interrupt (completion of judgment) because V_{DD} is lower than the specified threshold voltage (V_{VLS0}) after three samplings.
A VLS0 interrupt (completion of judgment) is output regardless of the judgment result of V_{DD}.
(4) "0" is automatically written to VLS0EN to disable VLS0.
(5) The voltage level detection flag (VLS0F) is read from the CPU to check the judgment result.

[Note]
During the single mode, the LSI cannot enter the STOP/STOP-D mode. When the single mode operation is completed (make sure that the VLS0EN bit is set to "0"), enable the STOP/STOP-D mode.

Figures 22-7 shows an operation timing diagram when the single mode 2 with sampling is selected.

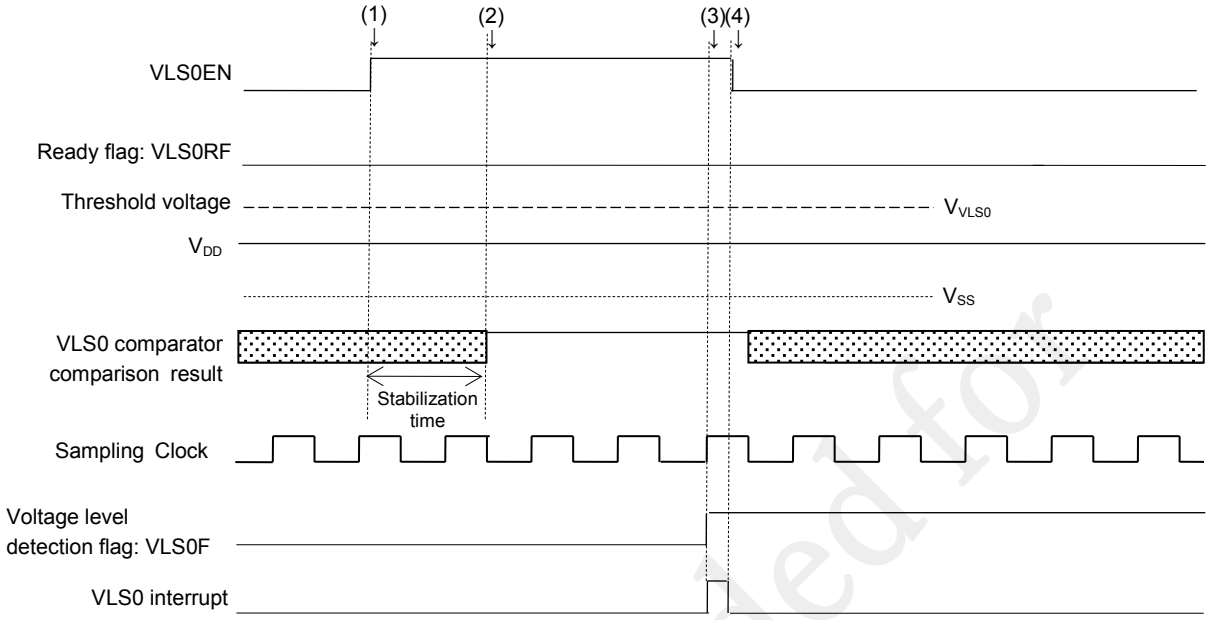


Figure 22-7 Operation Timing Diagram (when the single mode 2 with sampling is selected)

(1) "1" is written to VLS0EN from the CPU to enable VLS0.
(2) The VLS0 analog output is stabilized.
(3) The voltage level detection flag (VLS0F) is set to "1" to output a VLS0 interrupt (low voltage detection) because V_{DD} is lower than the specified threshold voltage (V_{VLS0}) after three samplings.
If V_{DD} is higher than the specified threshold voltage (V_{VLS0}), the voltage level detection flag (VLS0F) is set to "0" and a VLS0 interrupt (low voltage detection) is not output.
(4) "0" is automatically written to VLS0EN to disable VLS0.

[Note]
- When VLS0 is being stabilized, the LSI cannot enter the STOP/STOP-D mode. After starting the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode.
- When VLS0 is stopped (VLS0EN = "0") during the low voltage detection state (VLS0F = "1"), a VLS0 interrupt occurs.

Chapter 23 Successive Approximation Type A/D Converter

23. Successive Approximation Type A/D Converter

23.1 General Description

ML62Q1000 series has the Successive Approximation type A/D Converter (SA-ADC).
See Table 1-2 “Main Function List” in the Chapter 1 to confirm the presence/absence of function in each product.

23.1.1 Features

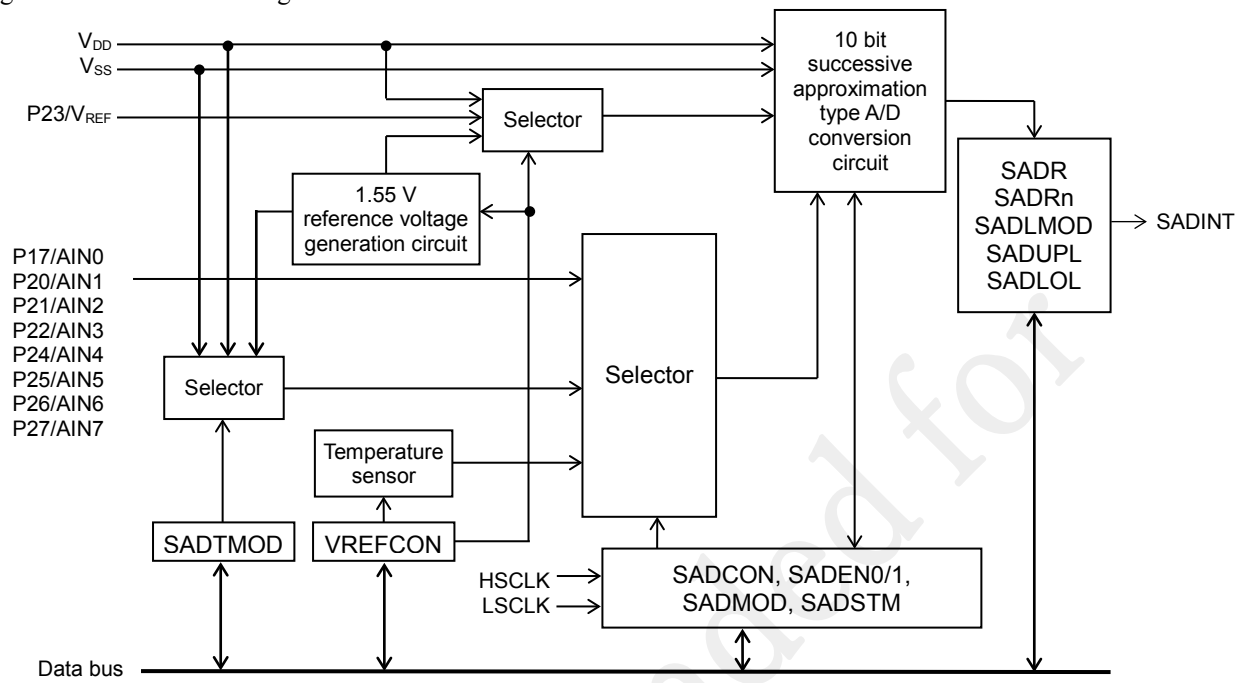
- Resolution: 10bit
- Conversion time: Min. 2.25μs/channel (conversion clock is 8MHz)
- Number of input channel: Max. 8ch

Package type	WQFN16 SSOP16	TSSOP20	WQFN24	TQFP32
Number of input channel	6ch		8ch	
Input pin name	AIN0~AIN3, AIN6, AIN7		AIN0~AIN7	

- Selectable reference voltage
  Voltage input from the V_{DD} pin, Internal reference voltage(approx.1.55V), External reference voltage(V_{REF} pin)
- Selectable sample time
- Scan conversion function (consecutive conversion for selected channels)
- Consecutive scan conversion with a specific interval time
- One conversion result register for each channel
- Upper /Lower limitation is configurable for the conversion result, generates an interrupt
- Built-in temperature sensor for the low-speed RC oscillation adjustment
- A/D converter self test function (full scale, zero scale, internal reference voltage)

23.1.2 Configuration

Figure 23-1 shows the configuration of SA-ADC.



SADCON : SA-ADC control register
SADEN0/1 : SA-ADC enable register
SADMOD : SA-ADC mode register
SADSTM : SA-ADC scan conversion interval setting register
SADR : SA-ADC result register
SADRn : SA-ADC result register n (n = 0 to 7, 16)
SADLMOD : SA-ADC upper/lower limit mode register
SADUPL : SA-ADC upper limit setting register
SADLOL : SA-ADC lower limit setting register
SADULSn : SA-ADC upper/lower limit status register n (n = 0 to 7, 16)
VREFCON : SA-ADC reference voltage control register
SADTMOD : SA-ADC test mode register

Figure 23-1 Configuration of SA-ADC

23.1.3 List of Pins

I/Os of the A/D converter are assigned to the secondary to octic functions of the genral port.
For details about pin assignment and how to set the secondary to octic functions of the genral port, see Chapter 17 "GPIO".

Pin name	I/O	Description
V_{DD}	-	Positive power supply pin for the successive approximation type A/D converter
V_{SS}	-	Negative power supply pin for the successive approximation type A/D converter
V_{REF}	I/O	Reference power supply pin for the successive approximation type A/D converter
AIN0	I	Successive approximation type A/D converter I/O 0
AIN1	I	Successive approximation type A/D converter I/O 1
AIN2	I	Successive approximation type A/D converter I/O 2
AIN3	I	Successive approximation type A/D converter I/O 3
AIN4	I	Successive approximation type A/D converter I/O 4
AIN5	I	Successive approximation type A/D converter I/O 5
AIN6	I	Successive approximation type A/D converter I/O 6
AIN7	I	Successive approximation type A/D converter I/O 7

[Note]
- When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to “0” as “Disable the input” and “Disable the output”, othewise a shoot-through current may flow.
- During the A/D conversion, influence of noise can be reduced by avoiding of switching other pins or by performing the A/D conversion in the HALT mode.

23.2 Description of Registers

23.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF800	SA-ADC result register 0	SADR0L	SADR0	R	8/16	0x00
0xF801		SADR0H		R	8	0x00
0xF802	SA-ADC result register 1	SADR1L	SADR1	R	8/16	0x00
0xF803		SADR1H		R	8	0x00
0xF804	SA-ADC result register 2	SADR2L	SADR2	R	8/16	0x00
0xF805		SADR2H		R	8	0x00
0xF806	SA-ADC result register 3	SADR3L	SADR3	R	8/16	0x00
0xF807		SADR3H		R	8	0x00
0xF808	SA-ADC result register 4	SADR4L	SADR4	R	8/16	0x00
0xF809		SADR4H		R	8	0x00
0xF80A	SA-ADC result register 5	SADR5L	SADR5	R	8/16	0x00
0xF80B		SADR5H		R	8	0x00
0xF80C	SA-ADC result register 6	SADR6L	SADR6	R	8/16	0x00
0xF80D		SADR6H		R	8	0x00
0xF80E	SA-ADC result register 7	SADR7L	SADR7	R	8/16	0x00
0xF80F		SADR7H		R	8	0x00
0xF810~ 0xF81F	Reserved register	-	-	R	8	0x00
0xF820	SA-ADC result register 16	SADR16L	SADR16	R	8/16	0x00
0xF821		SADR16H		R	8	0x00
0xF822	SA-ADC result register	SADRL	SADR	R	8/16	0x00
0xF823		SADRH		R	8	0x00
0xF824	SA-ADC upper/lower limit status register 0	SADULS0L	SADULS0	R/W	8/16	0x00
0xF825		SADULS0H		R/W	8	0x00
0xF826	SA-ADC upper/lower limit status register 1	SADULS1L	SADULS1	R/W	8/16	0x00
0xF827		SADULS1H		R/W	8	0x00
0xF828	SA-ADC mode register	SADMODL	SADMOD	R/W	8/16	0x00
0xF829		SADMODH		R/W	8	0x00
0xF82A	SA-ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF82B		SADCONH		R/W	8	0x00
0xF82C	SA-ADC enable register 0	SADEN0L	SADEN0	R/W	8/16	0x00
0xF82D		SADEN0H		R/W	8	0x00
0xF82E	SA-ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF82F		SADEN1H		R/W	8	0x00
0xF830	Reserved register	-	-	R	8	0x00
0xF831		-		R	8	0x00
0xF832	SA-ADC conversion interval register	SADSTML	SADSTM	R/W	8/16	0x00
0xF833		SADSTMH		R/W	8	0x00
0xF834	SA-ADC upper/lower limit mode register	SADLMODL	SADLMOD	R/W	8/16	0x00
0xF835		SADLMODH		R/W	8	0x00
0xF836	SA-ADC upper limit setting register	SADUPLL	SADUPL	R/W	8/16	0xC0
0xF837		SADUPLH		R/W	8	0xFF
0xF838	SA-ADC lower limit setting register	SADLOLL	SADLOL	R/W	8/16	0x00
0xF839		SADLOLH		R/W	8	0x00

0xF83A	Reference voltage control register	VREFCON	-	R/W	8	0x00
0xF83B	Reserved register	-	-	R	8	0x00
0xF83C	SA-ADC interrupt mode register	SADIMOD	-	R/W	8	0x00
0xF83D	Reserved register	-	-	R	8	0x00
0xF83E	SA-ADC trigger register	SADTRG	-	R/W	8	0x00
0xF83F	Reserved register	-	-	R	8	0x00
0xF0BA	SA-ADC test mode register	SADTMOD	-	R/W	8	0x00
0xF0BB	Reserved register	-	-	R	8	0x00

[Note]
Registers with a word symbol can be word accessed. For word access, specify an even address.

23.2.2 SA-ADC Result Register n (SADRn: n=0 to 7, 16)

Address: 0xF800, 0xF802, 0xF804, 0xF806, 0xF808, 0xF80A, 0xF80C, 0xF80E, 0xF820
Access: R
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADRn															
Byte symbol	SADRnH								SADRnL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
Access type	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

SADRn (n = 0 to 7, 16) is a special function register (SFR) used to store SA-ADC conversion results on channels 0 to 7 and channel 16 (temperature sensor).
The A/D conversion result of each channel can be read from SADRn.

Description of bits

- **d15 to d6** (bits 15-6)
  The 10bit A/D conversion result of channel n are stored to the d15 bit to d6 bit of SADRn(n= 0 to 7, 16).

Supported channels

Symbol name	Channel
SADR0	Conversion result of channel 0
SADR1	Conversion result of channel 1
SADR2	Conversion result of channel 2
SADR3	Conversion result of channel 3
SADR4	Conversion result of channel 4
SADR5	Conversion result of channel 5
SADR6	Conversion result of channel 6
SADR7	Conversion result of channel 7
SADR16	Conversion result of channel 16 (temperature sensor)

23.2.3 SA-ADC Result Register (SADR)

Address: 0xF822
Access: R
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADR															
Byte symbol	SADRH								SADRL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
Access type	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

SADR is a special function register (SFR) used to store SA-ADC conversion results on channels 0 to 7, 16 (temperature sensor) and 17 (A/D converter test function).
The A/D conversion result of all channels are stored to the SADR(The result of each channel is overwritten).
Use this SADR when transfering conversion results on multiple channels to RAM using DMA controller.
The A/D conversion test result on channel 17 is stored to the SADR.

Description of bits

- **d15 to d6** (bits 15-6)
  The 10bit A/D conversion result of channel n(n= 0 to 7, 16 and 17) are overwritten to the d15 bit to d6 bit.

Supported channels

Symbol name	Supported block
SADR	Conversion results of channels 0 to 7, 16 and 17

23.2.4 SA-ADC Upper/Lower Limit Status Register 0 (SADULS0)

Address: 0xF824
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADULS0															
Byte symbol	SADULS0H								SADULS0L							
Bit symbol	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	SAULS07	SAULS06	SAULS05	SAULS04	SAULS03	SAULS02	SAULS01	SAULS00
Access type	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

SAULS0 is a special function register (SFR) used to indicate whether the SA-ADC conversion result matches to the condition of upper/lower limit.

Description of bits

- **SAULS07 to SAULS00 (**bits 7-0)
  The SAULS07 to SAULS00 bits are used to indicate whether the A/D conversion results of channels 0 to 7 matches to the condition of upper/lower limit specified in SALMD[1:0] bits of SA-ADC upper/lower limit mode register(SADLMOD).
  The results are not updated when the SALEN bit is “0”.
  The corresponding bits get “1” if the condition matched and holds “1” until the bits are cleared or the LSI gets the system reset.
  When using the A/D conversion result upper/lower limit detection function (SALEN=1), the interrupt request is generated at the same time the corresponding bit gets “1”. Refer to Figure 23-5 for the timing of the interrupt and updates of detection result.
  SAULS07~SAULS00 is forcibly cleared to "0" by writing 1 to the bit. The writing “0” does not clear the bit.

SAULS00~07	Description
0	The A/D conversion result does not match to the condition of upper/lower limit (initial value)
1	The A/D conversion result matches to the condition of upper/lower limit

Supported channels

Symbol name	Channel
SAULS00	Detection result for the upper/lower limit on channel 0
SAULS01	Detection result for the upper/lower limit on channel 1
SAULS02	Detection result for the upper/lower limit on channel 2
SAULS03	Detection result for the upper/lower limit on channel 3
SAULS04	Detection result for the upper/lower limit on channel 4
SAULS05	Detection result for the upper/lower limit on channel 5
SAULS06	Detection result for the upper/lower limit on channel 6
SAULS07	Detection result for the upper/lower limit on channel 7

[Note]
- Do not use bit access instructions and use word or byte access instructions for writing the SADULS0 register.
- When using the A/D conversion result upper/lower limit detection function (SALEN=1), the interrupt can be cleared by clearing the corresponding bit of SAULS07~SAULS00 or by reseting the LSI.
- When performing the A/D conversion only one time, confirm the bit of SAULS07~SAULS00 is “0” before setting SARUN bit to “1”.
- When performing the consecutive A/D conversion (SALPEN=1), confirm the bit of SAULS07~SAULS00 is “0”, before the next A/D conversion ends.

23.2.5 SA-ADC Upper/Lower Limit Status Register 1 (SADULS1)

Address: 0xF826
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADULS1															
Byte symbol	SADULS1H								SADULS1L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SAULS16
Access type	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

SAULS1 is a special function register (SFR) used to indicate whether the SA-ADC conversion result matches to the condition of upper/lower limit.

Description of bits

- **SAULS16** (bit 0)
  The SAULS16 bit is used to indicate whether the A/D conversion result of channel 16 (temperature sensor) matches to the condition of upper/lower limit specified in SALMD[1:0] bits of SA-ADC upper/lower limit mode register(SADLMOD).
  The results are not updated when the SALEN bit is “0”.
  The corresponding bits get “1” if the condition matched and holds “1” until the bits are cleared or the LSI gets the system reset.
  When using the A/D conversion result upper/lower limit detection function (SALEN=1), the interrupt request is generated at the same time the corresponding bit gets “1”. Refer to Figure 23-5 for the timing of the interrupt and updates of detection result.
  SAULS16 is forcibly cleared to "0" by writing 1 to the bit. The writing “0” does not clear the bit.

SAULS16	Description
0	The A/D conversion result does not match to the condition of upper/lower limit (initial value)
1	The A/D conversion result matches to the condition of upper/lower limit

[Note]
- Do not use bit access instructions and use word or byte access instructions for writing the SADULS1 register.
- When using the A/D conversion result upper/lower limit detect function (SALEN=1), the interrupt can be cleared by clearing the corresponding bit of SAULS16 or by reseting the LSI.
- When performing the A/D conversion only one time, confirm the bit of SAULS16 is “0” before setting SARUN bit to “1”.
- When performing the consecutive A/D conversion (SALPEN=1), confirm the bit of SAULS16 is “0”, before the next A/D conversion ends.

23.2.6 SA-ADC Mode Register (SADMOD)

Address: 0xF828
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADMOD															
Byte symbol	SADMODH								SADMODL							
Bit symbol	-	-	-	-	-	-	-	SAINIT	SASHT3	SASHT2	SASHT1	SASHT0	SACK2	SACK1	SACK0	SALP
Access type	R	R	R	R	R	R	R	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

SADMOD is a special function register (SFR) used to control the operation of the SA-ADC.

Description of bits

- **SALP** (bit 0)
  The SALP bit is used to select whether A/D conversion is performed once only for each channel or consecutively. The conversion interval time in the consecutive A/D conversion mode is specified in the SA-ADC converion interval setting register (SADSTM).

SALP	Description
0	Single A/D conversion only (initial value)
1	Consecutive A/D conversion

- **SACK2 to SACK0** (bits 3-1)
  The SACK2 to SACK0 bits are used to select the SA-ADC operation clock.
  See Section 24.3.2 “A/D Conversion Time Setting” for the operating clock, A/D conversion time and sample time.

- **SASHT3 to SASHT0** (bits 7-4)
  The SASHT3 to SASHT0 bits are used to control the sample time.
  See Section 24.3.2 “A/D Conversion Time Setting” for details.

- **SAINIT** (bit 8)
  The SAINIT bit is used to control whether or not to discharge the electrical charge remained in the sample hold capacitor on the previous A/D conversion, before starting SA-ADC conversion.
  When the SAINIT bit is "1", the sample hold capacitance is discharged to Vss level.

SAINIT	Description
0	Starts A/D conversion without discharging the electrical charge accumulated in the sample hold capacitor (initial value)
1	Starts A/D conversion after discharging the electrical charge accumulated in the sample hold capacitor

[Note]
Write “0” to SADMODH[7:1]. Operation when “1” is written to the bits cannot be guranteed.

23.2.7 SA-ADC Control Register (SADCON)

Address: 0xF82A
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADCON															
Byte symbol	SADCONH								SADCONL							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SATGEN	SARUN
Access type	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

SADCON is a special function register (SFR) used to control the operation of the SA-ADC.

Description of bits

- **SARUN** (bit 0)
  The SARUN bit is used to start or stop SA-ADC conversion.
  Write "1" to this bit to start A/D conversion, and "0" to stop it.
  When "0" is written to SALP and then A/D conversion on the channel with the largest channel number among the selected ones is terminated, the SARUN bit is automatically set to "0".
  When "1" is written to SALP, A/D conversion is repeated until the software writes "0" to SARUN.

SARUN	Description
0	Stop conversion (initial value)
1	Start conversion

- **SATGEN** (bit 1)
  The SATGEN bit is used to control conversion started by the trigger event.

SATGEN	Description
0	Disable trigger operation (initial value)
1	Enable trigger operation

[Note]
- Start A/D conversion with one or more channels selected in the SA-ADC enable registers (SADEN0, SADEN1). If no channel is set, operation does not start.
- Enter STOP/STOP-D mode after checking SARUN bit is “0”, it does not enter the STOP/STOP-D mode when the SARUN bit is “1”.
- When SACK2~0 bits are set to 0x7, it takes max. 3 clocks of the low-speed clock(LSCLK) to start or stop the A/D conversion after setting or resetting the SARUN bit.

23.2.8 SA-ADC Enable Register 0 (SADEN0)

Address: 0xF82C
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADEN0															
Byte symbol	SADEN0H								SADEN0L							
Bit symbol	-	-	-	-	-	-	-	-	SACH07	SACH06	SACH05	SACH04	SACH03	SACH02	SACH01	SACH00
Access type	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

SADEN0 is a special function register (SFR) used to select A/D conversion channel(s).

Description of bits

- **SACH0n** (bits7-0)
  The SACH0(n=0~7) bits are used to select channels 0 to 7 to meke the A/D conversion.

SACHn	Description
0	Disable the conversion on channel n (initial value)
1	Enable the conversion on channel n

Supported channels

Symbol name	Channel
SACH00	Enable or Disable the A/D conversion on channel 0
SACH01	Enable or Disable the A/D conversion on channel 1
SACH02	Enable or Disable the A/D conversion on channel 2
SACH03	Enable or Disable the A/D conversion on channel 3
SACH04	Enable or Disable the A/D conversion on channel 4
SACH05	Enable or Disable the A/D conversion on channel 5
SACH06	Enable or Disable the A/D conversion on channel 6
SACH07	Enable or Disable the A/D conversion on channel 7

[Note]
- When multiple bits of SACHn are set to “1”, the A/D conversion starts in the order of smaller channel number.
- Do not start the A/D conversion when the all SACHn bits are "0". In that case the SARUN bit does not get to “1”.

23.2.9 SA-ADC Enable Register 1 (SADEN1)

Address: 0xF82E
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADEN1															
Byte symbol	SADEN1H								SADEN1L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SACH17	SACH16
Access type	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

SADEN1 is a special function register (SFR) used to select an A/D conversion channel.

Description of bits

- **SACHn** (bit 1 to 0)
  The SACHn bit (n=16, 17) are used to select channel 16(temperature sensor) and channel 17 (A/D converter test) to meke the A/D conversion.

SACHn	Description
0	Disable the conversion on channel n (initial value)
1	Enable the conversion on channel n

Supported channels

Symbol name	Channel
SACH16	Enable or Disable the A/D conversion on channel 16 (temperature sensor)
SACH17	Enable or Disable the A/D conversion on channel 17 (A/D converter test)

[Note]
- When multiple bits of SACHn are set to “1”, the A/D conversion starts in the order of smaller channel number.
- Do not start the A/D conversion when the all SACHn bits are "0". In that case the SARUN bit does not get to “1”.
- When using the channel 16 (SACH16), enable the intenal reference voltage/temperature sensor and select the internal reference voltage by setting the reference voltage control register (VREFCON).

23.2.10 SA-ADC Conversion Interval Setting Register (SADSTM)

Address: 0xF832
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADSTM															
Byte symbol	SADSTMH								SADSTML							
Bit symbol	SADSTM15	SADSTM14	SADSTM13	SADSTM12	SADSTM11	SADSTM10	SADSTM9	SADSTM8	SADSTM7	SADSTM6	SADSTM5	SADSTM4	SADSTM3	SADSTM2	SADSTM1	SADSTM0
Access type	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

SADSTM is a special function register (SFR) used to set the interval time in the consecutive A/D conversion mode. The interval time is determined by the PLL reference frequency(Code Opion) and the setting value of SACK2-0 bits(SADMOD) and SADSTM.

A/D conversion interval time = HSCLK 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 HSCLK.

[Note]
When SACK2-0 is “111”, the interval time is always minimum (0ns).

23.2.11 SA-ADC Upper/Lower Limit Mode Register (SADLMOD)

Address: 0xF834
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADLMOD															
Byte symbol	SADLMODH								SADLMODL							
Bit symbol	-	-	-	-	-	-	SALMD1	SALMD0	-	-	-	-	-	-	-	SALEN
Access type	R	R	R	R	R	R	R/W	R/W	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

SADLMOD is a special function register (SFR) used to set a mode in the A/D conversion result upper/lower limit detection function.

Description of bits

- **SALEN** (bit 0)

The SALEN bit is used to enable or disable the A/D conversion result upper/lower limit detection function. SA-ADC Upper/Lower Limit Status Register 0,1 (SADULS0,1) are not updated when the SALEN bit is "0".

SALEN	Description
0	Disable the upper/lower limit function for the A/D conversion (initial value)
1	Enable the upper/lower limit function for the A/D conversion

- **SALMD1 to SALMD0** (bits 9-8)

The SALMD1 to SALMD0 bits are used to set a condition of the A/D conversion result upper/lower limit detection. If the condition is satisfied, corresponding bits of the SA-ADC upper/lower status registers 0 and 1(SADULS0 and SADULS1) get to "1" and generates the SA-ADC intterupt request.

SALMD1	SALMD0	Description
0	0	Generates an interrupt when SADLOL ≦ A/D conversion value ≦ SADUPL (initial value)
0	1	Generates an interrupt when A/D conversion value > SADUPL
1	0	Generates an interrupt when A/D conversion value < SADLOL
1	1	Generates an interrupt when A/D conversion value > SADUPL or A/D conversion value < SADLOL

[Note]

- The upper/lower limit detection function is available to make the interrupt request for the A/D conversion result on all seleted channels.
- If the interrupt occurred by satisfying the upper/lower limit detection condition, check the SA-ADC upper/lower status registers 0 and 1(SADULS0 and SADULS1) to see which channel of A/D conversion result matched to the condition.

23.2.12 SA-ADC Upper Limit Setting Register (SADUPL)

Address: 0xF836
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFC0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADUPL															
Byte symbol	SADUPLH								SADUPLL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
Access type	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	R
Initial value	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

SADUPL is a special function register (SFR) used to set the upper limit of A/D conversion result.

23.2.13 SA-ADC Lower Limit Setting Register (SADLOL)

Address: 0xF838
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SADLOL															
Byte symbol	SADLOLH								SADLOLL							
Bit symbol	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
Access type	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	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SADLOL is a special function register (SFR) used to set the lower limit of A/D conversion result.

23.2.14 SA-ADC Reference Voltage Control Register (VREFCON)

Address: 0xF83A
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								VREFCON							
Bit symbol	-	-	-	-	-	-	-	-	-	-	VREFP1	VREFP0	-	-	-	VREFEN
Access type	R	R	R	R	R	R	R	R	R	R	R/W	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

VREFCON is a special function register (SFR) used to select the operation of the temperature sensor, V_{REF} pin, and internal reference voltage (1.55 V).

Description of bits

- **VREFEN** (bit 0)
  The VREFEN bit is used to enable the operation of internal reference voltage and the temperature sensor.
  When using the internal reference voltage(approx. 1.55V) or temperature sensor, set the VREFEN bit to "1".

VREFEN	Description
0	Disable the operation of internal reference voltage and temperature sensor (initial value)
1	Enable the operation of internal reference voltage and temperature sensor

- **VREFP1 to VREFP0** (bits 5-4)
  The VREFP1 to VREFP0 bits are used to select the reference voltage for the A/D conversion.

VREFP1	VREFP0	Description
0	0	Voltage input from the V_{DD} pin (initial value)
0	1	Voltage input from the V_{REF} pin
1	0	Voltage generated by the internal reference voltage circuit (1.55V)
1	1	Do not use (Voltage input from the V_{DD} pin)

[Note]
- It takes 200us(Max.) until the internal reference voltage gets stable after setting VREFEN bit to "1". Start the A/D conversion after waiting the stabilization time.
- The internal reference voltage(Approx. 1.55V) can be output to the general port(P23) by setting the VREFEN bit to"1" and setting 0x70 to P2MOD3 register. However in that case, it is possbible to get incorrect A/D conversion results as affected by external factors.
- When using the external reference voltage input from V_{REF} pin(P23), set VREFP1 bit to "0" and VREFP0 bit to "1" and P2MOD3 register to 0x00.

23.2.15 SA-ADC Interrupt Mode Register (SADIMOD)

Address: 0xF83C
Access: R/W
Access size: 8 Bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								SADIMOD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADIMD
Access type	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

SADIMOD is a special function register (SFR) used to control the interrupt mode of the SA-ADC.

Description of bits

- **SADIMD (bit 0)**
  The SADIMD bit is used to control the timing of SA-ADC interrupt(SADINT) occurrence.

SADIMD	Description
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

[Note]
If SALEN bit of the SA-ADC upper/lower limit mode register (SADLMOD) is set to “1”, the interrupt from the upper/lower limit detection function gets enabled and the setting for the SADIMD bit of SA-ADC Interrupt Mode Register (SADIMOD) is invalid.

23.2.16 SA-ADC Trigger Register (SADTRG)

Address: 0F83EH
Access: R/W
Access size: 8 Bits
Initial value: 00H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								SADTRG							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	SASTS4	SASTS3	SASTS2	SASTS1	SASTS0
Access type	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

SADMOD is a special function register (SFR) used to control the operation of the SA-ADC.

Description of bits

- **SASTS4 to SASTS0** (bits 4-0)
  The SASTS4 to SASTS0 bits are used to select the source of the trigger event for SA-ADC.

SASTS*					Description
4	3	2	1	0	
0	0	0	0	0	TM0INT :16bit Timer 0 interrupt (initial value)
0	0	0	0	1	TM1INT :16bit Timer 1 interrupt
0	0	0	1	0	Reserved
0	0	0	1	1	Reserved
0	0	1	0	0	Reserved
0	0	1	0	1	Reserved
0	0	1	1	0	Reserved
0	0	1	1	1	Reserved
0	1	0	0	0	LTBC0INT :Low speed Time Base Counter 0 interrupt
0	1	0	0	1	Do not use
:	:	:	:	:	
1	1	1	1	1	

[Note]
When selecting the 16bit Timer interrupts (TM0INT and TM1INT), set the THn8BM bit of the 16bit Timer n Mode Register (TMHnMOD) to "0" to select the 16bit timer mode.

23.2.17 SA-ADC test mode register (SADTMOD)

Address: 0xF0BA
Access: R/W
Access size: 8 Bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								SADTMOD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADTMD1	SADTMD0
Access type	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

SADTMOD is a register (SFR) used to control the successive approximation type A/D converter test function.
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 (about 1.55 V) . The A/D conversion result is stored in the SA-ADC result register (SADR).

Description of bits

- **SADTMD1 to SADTMD0** (bits 1-0)
  The SADTMD1 to SADTMD0 bits are used to set the successive approximation type A/D converter test function.

SADTMD1	SADTMD0	Description
0	0	Successive approximation type A/D converter test function is not used (initial value)
0	1	Successive approximation type A/D converter full scale test function (AIN = V_{DD}/$V_{REF/}$ Internal reference voltage) is used
1	0	Successive approximation type A/D converter zero scale test function (AIN = V_{SS}) is used
1	1	Successive approximation type A/D converter internal reference test function (AIN = Internal reference voltage) is used

Follow this procedure to check if the successive approximation type A/D converter works correctly.

(1) Enable the operation of internal reference voltage and select VDD as the reference for A/D conversion (VREFCON = 0x01)
(2) A/D convert AINn pin. (conversion result 1)
(3) A/D convert AIN=full scale by setting the SADTMOD register (SADTMOD=0x01).
(4) A/D convert the AINn pin. (conversion result 2)
(5) A/D convert AIN=zero scale by setting the SADTMOD register (SADTMOD=0x02).
(6) A/D convert the AINn pin. (conversion result 3)
(7) A/D convert AIN=internal reference voltage by setting the SADTMOD register (SADTMOD=0x03).
(8) A/D convert the AINn pin. (conversion result 4)
(9) Confirm conversion result 1 = conversion result 2 = conversion result 3 = conversion result 4. Use the same AINn pin for the A/D conversion in (2), (4), (6) and (8).
(10) Confirm the conversion result in (3), (5) and (7) is different each other and also different from the result in (2), (4), (6) and (8).

23.3 Description of Operation

23.3.1 Operation of Successive Approximation Type A/D Converter

Figure 23-2 shows a setting example for performing the A/D conversion only once using channel 1 and 0 and Figures 23-3 and 23-4 show the operation waveform.

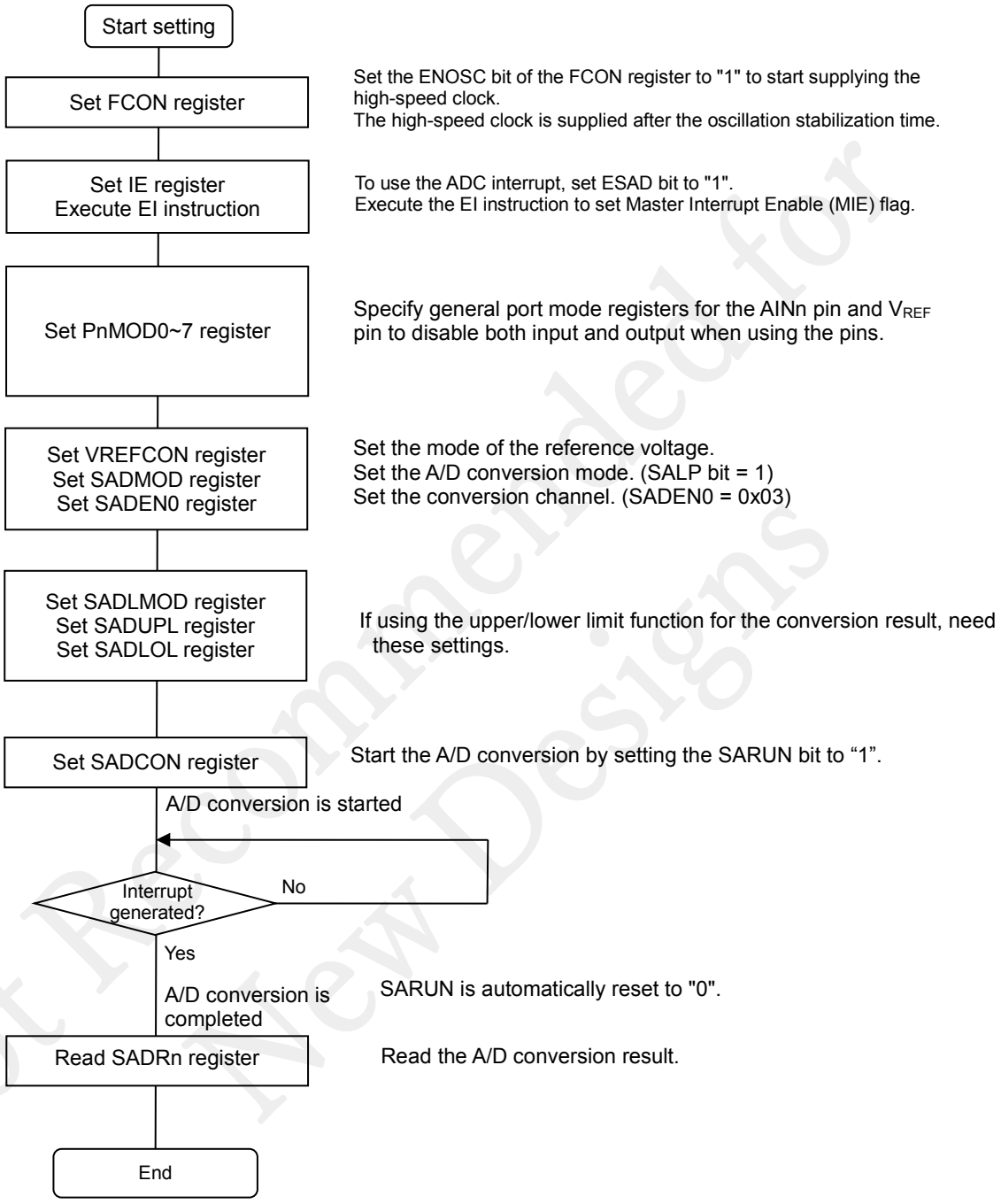
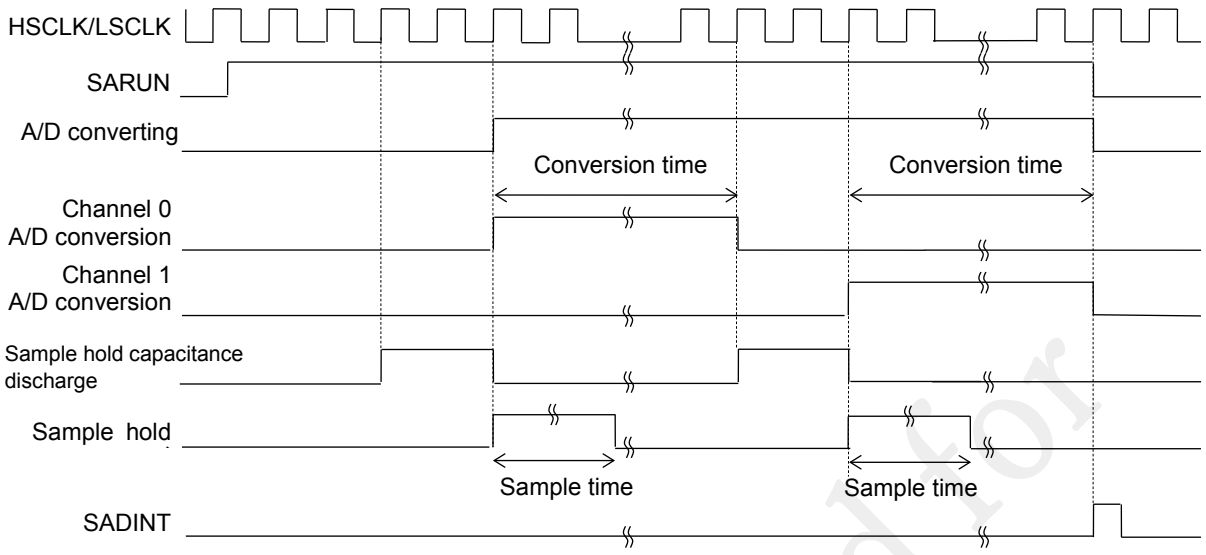


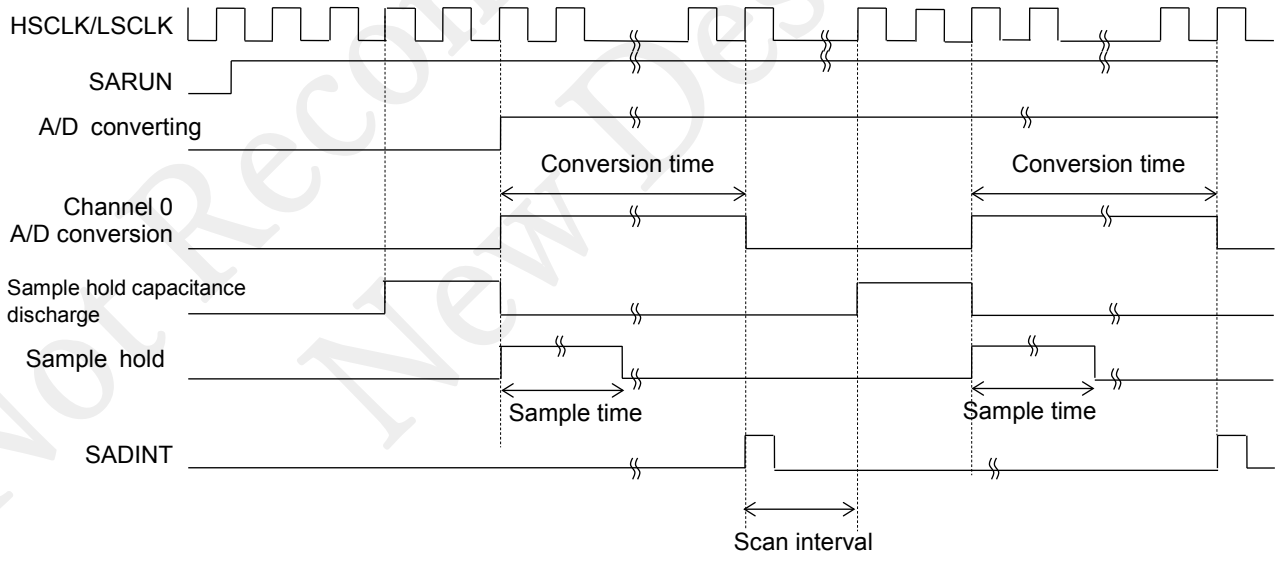
Figure 23-2 Setting Example of SA-ADC



With/without discharge is selectable by the SAINIT bit. It takes two cycles of SAD_CLK to discharge the sample hold capacitance.
The sample time is selectable by the SASHT1 and SASHT0 bits.
The conversion time is selectable by the SACK1 and SACK0 bits.

Figure 23-3 Operation Waveform of SA-ADC (One Conversion)

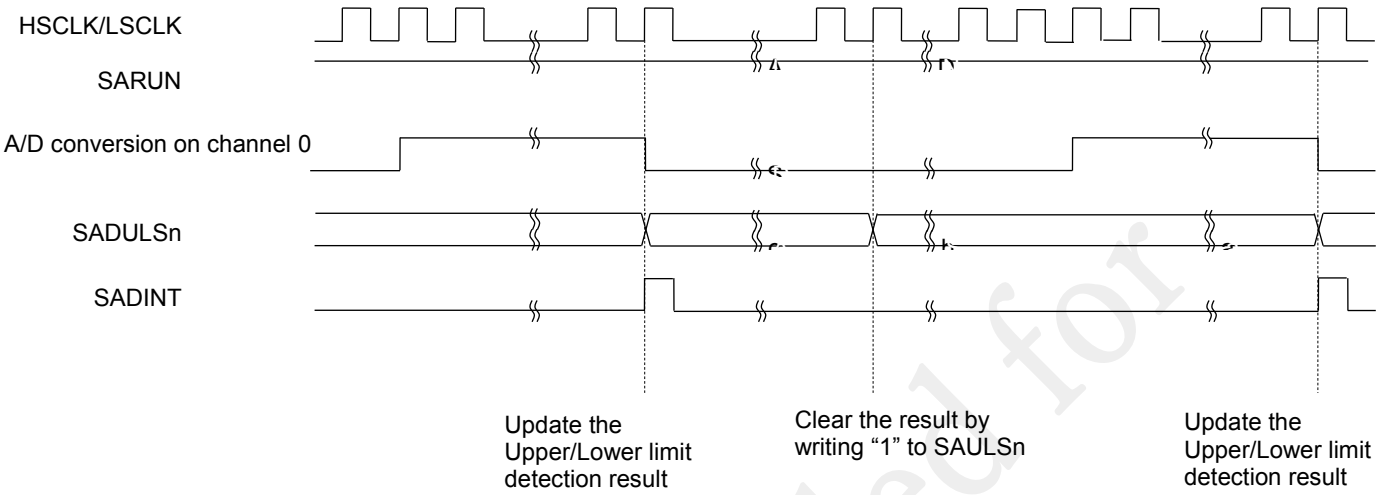
Figure 23-4 shows the operation waveform when the continuous A/D conversion is performed using channel 0.



With/without discharge is selectable by the SAINIT bit. It takes two cycles of SAD_CLK to discharge the sample hold capacitance.
The sample time is selectable by the SASHT1 and SASHT0 bits.
The conversion time is selectable by the SACK1 and SACK0 bits.
The scan interval is configurable by the SADSTM register.

Figure 23-4 Operation Waveform of SA-ADC (Continuous Conversion)

Figure 23-5 shows the timing chart of when the conversion result matches the upper/lower limit condition set by the SALMD1~0 bits.



It takes max. one clock of SAD_CLK to clear the uppler/lower limit detection result after writing the SAULSn bit to “1”.

Figure 23-5 Timing chart of SADULSn when using the Update the Upper/Lower limit detection function

23.3.2 A/D Conversion Time Setting

Available A/D conversion time and sample time are different depending on following condition of configurations.

- Reference voltage selected in the SA-ADC Reference Voltage Control Register (VREFCON)
- PLL oscillation frequency selected in the Code Option 1 Registe (CODEOP1)
- A/D conversion operating clock (SAD_CLK) selected in the SA-ADC Mode Register (SADMOD)

Table 23-1 ~ 23-6 on page 23-25 ~ 23-27 show relation between the setting condition of SADMOD and the A/D conversion time.
Table 23-7 ~ 23-12 on page 23-28 ~ 23-30 show relation between the setting condition of SADMOD and the sample time.
Refer to different tables for different reference voltage mode.

Reference voltage	Time	PLL oscillation frequency		
		16MHz	24MHz	32MHz
V_{DD} pin or External V_{REF} pin	A/D conversion time	Table 23-1	Table 23-3	Table 23-5
	Sample time	Table 23-7	Table 23-9	Table 23-11
Internal reference voltage	A/D conversion time	Table 23-2	Table 23-4	Table 23-6
	Sample time	Table 23-8	Table 23-10	Table 23-12

Table 23-1 A/D conversion time when using V_{DD} or V_{REF} pin for reference
(PLL oscillation frequency is 16MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs*1	Do not use	Do not use	28µs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			30µs	15µs			
0010				32µs	16µs	8.0µs		
0011				34µs	17µs	8.5µs	4.25µs	
0100				36µs	18µs	9.0µs	4.50µs	2.25µs
0101				38µs	19µs	9.5µs	4.75µs	2.38µs
0110				40µs	20µs	10.0µs	5.00µs	2.50µs
0111				42µs	21µs	10.5µs	5.25µs	2.63µs
1000				58µs	29µs	14.5µs	7.25µs	3.63µs
1001				90µs	45µs	22.5µs	11.25µs	5.63µs
1010				122µs	61µs	30.5µs	15.25µs	7.63µs
1011				154µs	77µs	38.5µs	19.25µs	9.63µs
1100				186µs	93µs	46.5µs	23.25µs	11.63µs
1101				218µs	109µs	54.5µs	27.25µs	13.63µs
1110				250µs	125µs	62.5µs	31.25µs	15.63µs
1111				282µs	141µs	70.5µs	35.25µs	17.63µs

*1 LSCLK(32.768kHz) is selected.

Table 23-2 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 16MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs*1	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				58µs				
1001				90µs	45µs			
1010				122µs	61µs			
1011				154µs	77µs	38.5µs		
1100				186µs	93µs	46.5µs		
1101				218µs	109µs	54.5µs		
1110				250µs	125µs	62.5µs		
1111				282µs	141µs	70.5µs	35.25µs	

*1 LSCLK(32.768kHz) is selected.

Table 23-3 A/D conversion time when using V_{DD} or V_{REF} pin for reference
(PLL oscillation frequency is 24MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs[*1]	Do not use	Do not use	37.34µs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			40.00µs	20.00µs			
0010				42.67µs	21.34µs	10.67µs		
0011				45.34µs	22.67µs	11.34µs	5.67µs	
0100				48.00µs	24.00µs	12.00µs	6.00µs	3.00µs
0101				50.67µs	25.34µs	12.67µs	6.34µs	3.17µs
0110				53.34µs	26.67µs	13.34µs	6.67µs	3.34µs
0111				56.00µs	28.00µs	14.00µs	7.00µs	3.50µs
1000				77.34µs	38.67µs	19.34µs	9.67µs	4.84µs
1001				120.00µs	60.00µs	30.00µs	15.00µs	7.50µs
1010				162.67µs	81.34µs	40.67µs	20.34µs	10.17µs
1011				205.34µs	102.67µs	51.34µs	25.67µs	12.84µs
1100				248.00µs	124.00µs	62.00µs	31.00µs	15.50µs
1101				290.67µs	145.34µs	72.67µs	36.34µs	18.17µs
1110				333.34µs	166.67µs	83.34µs	41.67µs	20.84µs
1111				376.00µs	188.00µs	94.00µs	47.00µs	23.50µs

[*1] LSCLK(32.768kHz) is selected.

Table 23-4 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 24MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs[*1]	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				77.34µs				
1001				120.00µs	60.00µs			
1010				162.67µs	81.34µs			
1011				205.34µs	102.67µs	51.34µs		
1100				248.00µs	124.00µs	62.00µs		
1101				290.67µs	145.34µs	72.67µs		
1110				333.34µs	166.67µs	83.34µs		
1111				376.00µs	188.00µs	94.00µs	47.00µs	

[*1] LSCLK(32.768kHz) is selected.

Table 23-5 A/D conversion time when using V_{DD} or V_{REF} pin for reference
(PLL oscillation frequency is 32MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs[*1]	Do not use	Do not use	21.00µs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			22.50µs	11.25µs			
0010				24.00µs	12.00µs	6.00µs		
0011				25.50µs	12.75µs	6.38µs	3.19µs	
0100				27.00µs	13.50µs	6.75µs	3.38µs	
0101				28.50µs	14.25µs	7.13µs	3.57µs	
0110				30.00µs	15.00µs	7.50µs	3.75µs	
0111				31.50µs	15.75µs	7.88µs	3.94µs	
1000				43.50µs	21.75µs	10.88µs	5.44µs	
1001				67.50µs	33.75µs	16.88µs	8.44µs	
1010				91.50µs	45.75µs	22.88µs	11.44µs	
1011				115.50µs	57.75µs	28.88µs	14.44µs	
1100				139.50µs	69.75µs	34.88µs	17.44µs	
1101				163.50µs	81.75µs	40.88µs	20.44µs	
1110				187.50µs	93.75µs	46.88µs	23.44µs	
1111				211.50µs	105.75µs	52.88µs	26.44µs	

[*1] LSCLK(32.768kHz) is selected.

Table 23-6 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 32MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	427µs[*1]	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				43.50µs				
1001				67.50µs	33.75µs			
1010				91.50µs	45.75µs			
1011				115.50µs	57.75µs	28.88µs		
1100				139.50µs	69.75µs	34.88µs		
1101				163.50µs	81.75µs	40.88µs		
1110				187.50µs	93.75µs	46.88µs		
1111				211.50µs	105.75µs	52.88µs	26.44µs	

[*1] LSCLK(32.768kHz) is selected.

[Note]
- When the A/D conversion mode with “discharging the electrical charge remained in the sample hold capacitor to Vss” is selected (SAINIT=1), the A/D conversion time gets max. 15% longer.
- The A/D conversion time does not include error of clock frequency.

Table 23-7 Sample time when using VDD or VREF pin for reference
(PLL oscillation frequency is 16MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50µs*1	Do not use	Do not use	2.00µs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			4.00µs	2.00µs			
0010				6.00µs	3.00µs	1.50µs		
0011				8.00µs	4.00µs	2.00µs	1.00µs	
0100				10.00µs	5.00µs	2.50µs	1.25µs	0.63µs
0101				12.00µs	6.00µs	3.00µs	1.50µs	0.75µs
0110				14.00µs	7.00µs	3.50µs	1.75µs	0.88µs
0111				16.00µs	8.00µs	4.00µs	2.00µs	1.00µs
1000				32.00µs	16.00µs	8.00µs	4.00µs	2.00µs
1001				64.00µs	32.00µs	16.00µs	8.00µs	4.00µs
1010				96.00µs	48.00µs	24.00µs	12.00µs	6.00µs
1011				128.00µs	64.00µs	32.00µs	16.00µs	8.00µs
1100				160.00µs	80.00µs	40.00µs	20.00µs	10.00µs
1101				192.00µs	96.00µs	48.00µs	24.00µs	12.00µs
1110				224.00µs	112.00µs	56.00µs	28.00µs	14.00µs
1111				256.00µs	128.00µs	64.00µs	32.00µs	16.00µs

*1 LSCLK(32.768kHz) is selected.

Table 23-8 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 16MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50µs*1	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				32.00µs				
1001				64.00µs	32.00µs			
1010				96.00µs	48.00µs			
1011				128.00µs	64.00µs	32.00µs		
1100				160.00µs	80.00µs	40.00µs		
1101				192.00µs	96.00µs	48.00µs		
1110				224.00µs	112.00µs	56.00µs		
1111				256.00µs	128.00µs	64.00µs	32.00µs	

*1 LSCLK(32.768kHz) is selected.

Table 23-9 Sample time when using VDD or VREF pin for reference
(PLL oscillation frequency is 24MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50μs*1	Do not use	Do not use	2.67μs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			5.33μs	2.67μs			
0010				8.00μs	4.00μs	2.00μs		
0011				10.67μs	5.33μs	2.67μs	1.33μs	
0100				13.33μs	6.67μs	3.33μs	1.67μs	0.83μs
0101				16.00μs	8.00μs	4.00μs	2.00μs	1.00μs
0110				18.67μs	9.33μs	4.67μs	2.33μs	1.17μs
0111				21.33μs	10.67μs	5.33μs	2.67μs	1.33μs
1000				42.67μs	21.34μs	10.67μs	5.34μs	2.67μs
1001				85.33μs	42.67μs	21.33μs	10.67μs	5.33μs
1010				128.00μs	64.01μs	32.00μs	16.01μs	8.00μs
1011				170.67μs	85.34μs	42.67μs	21.34μs	10.67μs
1100				213.33μs	106.67μs	53.33μs	26.67μs	13.33μs
1101				256.00μs	128.01μs	64.00μs	32.01μs	16.00μs
1110				298.67μs	149.34μs	74.67μs	37.34μs	18.67μs
1111				341.33μs	170.67μs	85.33μs	42.67μs	21.33μs

*1 LSCLK(32.768kHz) is selected.

Table 23-10 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 24MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50μs*1	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				42.67μs				
1001				85.33μs	42.67μs			
1010				128.00μs	64.01μs			
1011				170.67μs	85.34μs	42.67μs		
1100				213.33μs	106.67μs	53.33μs		
1101				256.00μs	128.01μs	64.00μs		
1110				298.67μs	149.34μs	74.67μs		
1111				341.33μs	170.67μs	85.33μs	42.67μs	

*1 LSCLK(32.768kHz) is selected.

Table 23-11 Sample time when using VDD or VREF pin for reference
(PLL oscillation frequency is 32MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50µs*1	Do not use	Do not use	1.50µs	Do not use	Do not use	Do not use	Do not use
0001	Do not use			3.00µs	1.50µs			
0010				4.50µs	2.25µs	1.13µs		
0011				6.00µs	3.00µs	1.50µs	0.75µs	
0100				7.50µs	3.75µs	1.88µs	0.94µs	
0101				9.00µs	4.50µs	2.25µs	1.13µs	
0110				10.50µs	5.25µs	2.63µs	1.31µs	
0111				12.00µs	6.00µs	3.00µs	1.50µs	
1000				24.00µs	12.00µs	6.00µs	3.00µs	
1001				48.00µs	24.00µs	12.00µs	6.00µs	
1010				72.00µs	36.00µs	18.00µs	9.00µs	
1011				96.00µs	48.00µs	24.00µs	12.00µs	
1100				120.00µs	60.00µs	30.00µs	15.00µs	
1101				144.00µs	72.00µs	36.00µs	18.00µs	
1110				168.00µs	84.00µs	42.00µs	21.00µs	
1111				192.00µs	96.00µs	48.00µs	24.00µs	

*1 LSCLK(32.768kHz) is selected.

Table 23-12 A/D conversion time when using the Internal reference
(PLL oscillation frequency is 32MHz)

SASHT3~0	SACK2~0							
	111	110	101	100	011	010	001	000
0000	30.50µs*1	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use	Do not use
0001	Do not use							
0010								
0011								
0100								
0101								
0110								
0111								
1000				24.00µs				
1001				48.00µs	24.00µs			
1010				72.00µs	36.00µs			
1011				96.00µs	48.00µs	24.00µs		
1100				120.00µs	60.00µs	30.00µs		
1101				144.00µs	72.00µs	36.00µs		
1110				168.00µs	84.00µs	42.00µs		
1111				192.00µs	96.00µs	48.00µs	24.00µs	

*1 LSCLK(32.768kHz) is selected.

Chapter 24 Regulator

24. Regulator

24.1 General Description

ML62Q1000 series has the regulator that generates the internal voltage (V_{DDL}), supplied to the internal logic circuits, flash memory, RAM, oscillation circuits and etc.
Connect the V_{DDL} pin to V_{SS} via a capacitor (1uF) to stabilize the regulator output voltage.

24.1.1 Features

- Low internal voltage (V_{DDL}) for operating the internal logic circuits
- The V_{DDL} goes down to approx. 1.1V to reduce the current in STOP-D mode, remaining data in the RAM and SFRs).

24.1.2 Configuration

Figure 24-1 shows the configuration of the regulator.

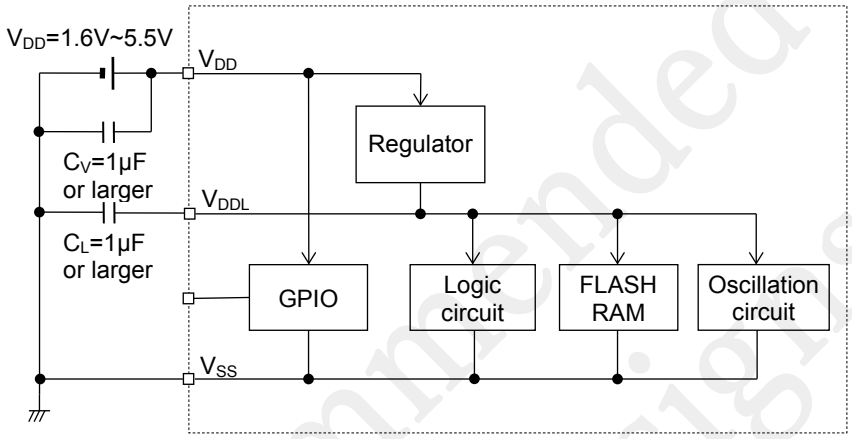


Figure 24-1 Regulator Configuration

24.1.3 List of Pins

Connect the V_{DDL} pin to VSS via a capacitor (1 µF) to stabilize the regulator output voltage.

Pin name	I/O	Function
V_{DDL}	-	Positive power supply pin for the internal logic circuits

[Note]
- For increasing the noise resistance, place the bypass capacitor (Cv 1µF) and the capacitor for V_{DDL} close to the power pins and keep the traces from capacitor to the pins as short as possible without going via through holes.
- The voltage for internal logic (V_{DDL}) is not capable for operating any external device.

24.2 Description of Operation

After power-on, the V_{DDL} voltage becomes approximately 1.55 V.
The regulator keepds the 1.55 V in CPU run mode, HALT mode, HALT-H mode and STOP modes.
The voltage becomes approximately 1.1 V in the STOP-D mode, in which the data of RAM and SFRs are retained.
Figure 24-2 shows the operation waveforms of the regulator.
For details about the STOP-D mode and the oscillation stabilization time, see Chapter 4 “Power Management” and Chapter 6 “Clock Generation Circuit”.

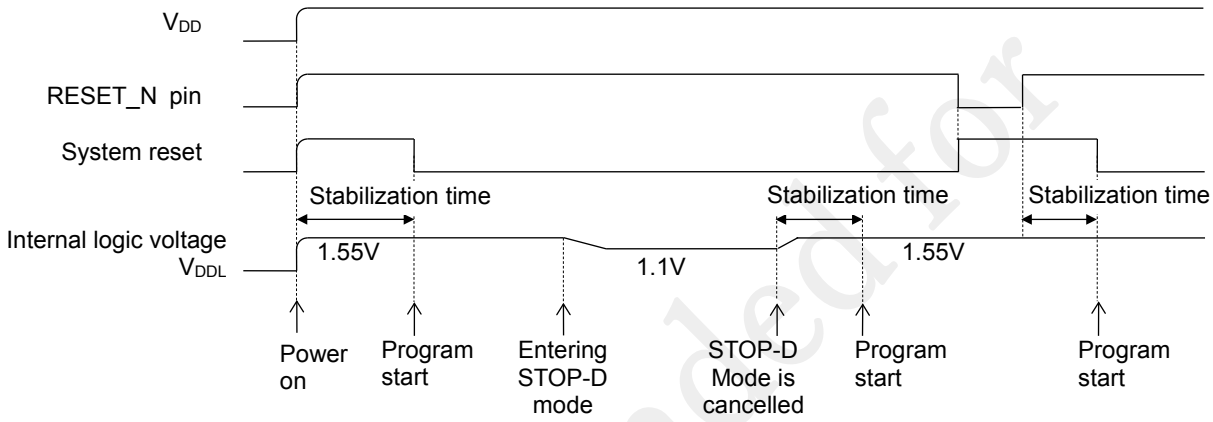


Figure 24-2 Regulator Operation Waveforms

24.2.1 Reference Voltage Output

See Section 23.2.14 “SA-ADC Reference Voltage Control Register (VREFCON)” for reference voltage output.

Chapter 25 Flash Memory

25. Flash Memory

25.1 General Description

ML62Q1000 series 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.

− Programming by the on-chip debug function
The flash memory in the program memory space and data flash area is programmable by using the on-chip debug emulator(EASE1000) or other flash programmers. See Chapter 28 “On-chip Debug Function" for details about the on-chip debug function.

− Self-Programming by using the specific function register(SFR)
The flash memory in the program memory space and data flash area is programmable by using the specific function registers(SFRs) controlled in the user application. See “Section 25.3 Self-programming" for details on how to operate the self-programming.

− Programming by the ISP (In-System Programming) function
The flash memory in the program memory space and data flash area is programmable by using the ISP function. See “Section 25.4 ISP function” for details on how to use the ISP function.

25.1.1 Features

- Program memory space and Data flash area Overview (Size and Address)

Part number	Program memory space		Data flash area	
	Size	Address	Size	Address
ML62Q1223A / 1233A	16K byte	0:0000 ~ 0:3FFF	2K byte (128 byte x 16 sector)	1F:0000 ~ 1F:07FF
ML62Q1224A / 1234A	24K byte	0:0000 ~ 0:5FFF		
ML62Q1225A / 1235A	32K byte	0:0000 ~ 0:7FFF		
ML62Q1245A / 1265A				
ML62Q1246A / 1266A	48K byte	0:0000 ~ 0:BFFF		
ML62Q1247A / 1267A	64K byte	0:0000 ~ 0:FFFF		

- Program memory space and Data flash area Overview (Functions and Characteristics)

Item		Program memory space	Data flash area
Erasing and programming unit	Chip erase(ISP only)	All area	All area
	Block erase	16K byte	2K byte
	Sector erase	1K byte	128 byte
	Programming	4 byte (32bit)	1 byte (8bit)
Erasing and programming time	Chip erase(ISP only)	Max. 85ms	Max. 85ms
	Block erase		
	Sector erase		
	Programming	Max. 80μs	Max. 40μs
Programming cycle		100 times	10,000 times
Erasing and programming temparature		0°C ~ 40°C	-40°C ~ 85°C
Background operation(BGO) function		−	Yes

25.2 Description of Registers

25.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
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	Flash data register 1	FLASHD1L	FLASHD1	R/W	8/16	0xFF
0xF095		FLASHD1H		R/W	8	0xFF
0xF096	Flash control register	FLASHCON	-	W	8	0x00
0xF097	Reserved register	-	-	W	8	0x00
0xF098	Flash acceptor	FLASHACP	-	W	8	0x00
0xF099	Reserved register	-	-	W	8	0x00
0xF09A	Flash segment register	FLASHSEG	-	R/W	8	0x00
0xF09B	Reserved register	-	-	R	8	0x00
0xF09C	Flash self register	FLASHSLF	-	R/W	8	0x00
0xF09D	Reserved register	-	-	R	8	0x00
0xF09E	Flash status register	FLASHSTA	-	R	8	0x00
0xF09F	Reserved register	-	-	R	8	0x00

[Note]
Word access is available for the registers having word symbols. To perform a word access, specify an even address.

25.2.2 Flash Address Register (FLASHA)

Address: 0xF090
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FLASHA															
Byte symbol	FLASHAH								FLASHAL							
Bit symbol	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0
Access type	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

FLASHA is a special function register (SFR) used to set the flash memory rewrite addresses.

Description of bits

- **FA15 to FA0** (bits 15-0)
  The FA15 to FA0 bits are used to set the address for erasing or rewrite.

[Note]
Note that the rewrite operation for the program memory space is performed by the unit of 4 bytes. In this case, the setting values of bits 1 and 0 are ignored.

25.2.3 Flash Segment Register (FLASHSEG)

Address: 0xF09A
Access: R/W
Access size: 8 bits
Initial value: 0x10

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FLASHSEG							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	FSEG4	FSEG3	FSEG2	FSEG1	FSEG0
Access type	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

FLASHSEG is a special function register (SFR) used to set the flash memory rewrite segment address.
Table 25-1 and Table 25-2 shows the address setting values for block erase and sector erase.

Description of bits

- **FSEG4 to FSEG0** (bits 4-0)
  The FSEG4 to FSEG0 bits are used to specify the flash memory segment address.
  For details, see the description of the flash address register (FLASHA).

Table 25-1 Address Setting Values for Block Erase

Segment	Block	Address	Size	FLASHSEG	FLASHA
Segment0	Block0	0x0000~0x3FFF	16K byte	0x00	0x0000
	Block1	0x4000~0x7FFF	16K byte		0x4000
	Block2	0x8000~0xBFFF	16K byte		0x8000
	Block3	0xC000~0xFFFF	16K byte		0xC000
Segment31	Block0	0x0000~0x07FF	2K byte	0x1F	0x0000

Table 25-2 Address Setting Values for Sector Erase

Segment	Block	Address	Size	FLASHSEG	FLASHA
Segment0	Sector0	0x0000~0x03FF	1K byte	0x00	0x0000
	Sector1	0x0400~0x07FF	1K byte		0x0400
	Sector2	0x0800~0x0BFF	1K byte		0x0800
	Sector3	0x0C00~0x0FFF	1K byte		0x0C00
	Sector4	0x1000~0x13FF	1K byte		0x1000
	:	:	:		:
	Sector63	0xFC00~0xFFFF	1K byte		0xFC00
Segment31	Sector0	0x0000~0x007F	128 byte	0x1F	0x0000
	Sector1	0x0080~0x00FF	128 byte		0x0080
	Sector2	0x0100~0x017F	128 byte		0x0100
	Sector3	0x0180~0x01FF	128 byte		0x0180
	Sector4	0x0200~0x027F	128 byte		0x0200
	:	:	:		:
	Sector15	0x0780～0x07FF	128 byte		0x0780

25.2.4 Flash Data Register 0 (FLASHD0)

Address: 0xF092
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FLASHD0															
Byte symbol	FLASHD0H								FLASHD0L							
Bit symbol	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
Access type	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

FLASHD0 is a special function register (SFR) used to set the flash memory rewrite data.
Use a four bytes specified in the FLASHD0L, FLASHD0H, FLASHD1L and FLASHD1H as the rewrite data to the program flash memory.
Use only one byte specified in the FLASHD0L to rewrite data in the data flash memory area. Writing data to the FLASHD0L starts programming the data flash memory. Data written to the FLASHD0H is invalid for programming the data falsh memory.

Description of bits

- **FD7 to FD0** (bits 7-0)
  The FD7 to FD0 bits are used to set the first byte of data.

- **FD15 to FD8** (bits 15-8)
  The FD15 to FD8 bits are used to set the second byte of data.

[Note]
- Specify a segment address to the FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.
- Back Ground Operation(BGO) function allows CPU continue running the program codes while programming the data flash memory. Confirm the end of programming by checking FDPRSTA bit of Flash Status Register(FLASHSTA).
- Erase data in the addresses in advance. Programmed data without erase cannot be guranteed.
- Do not access to unused area, otherwise the CPU may work incorrectly.

25.2.5 Flash Data Register 1 (FLASHD1)

Address: 0xF094
Access: R/W
Access size: 8/16 bits
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	FLASHD1															
Byte symbol	FLASHD1H								FLASHD1L							
Bit symbol	FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
Access type	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

FLASHD1 is a special function register (SFR) used to set the flash memory rewrite data.
Use a four bytes specified in the FLASHD0L, FLASHD0H, FLASHD1L and FLASHD1H as the rewrite data to the program flash memory.
Writing data to the FLASHD1H starts programming the program flash memory. Specify the data in the order of FLASHD0L, FLASHD0H, FLASHD1L and FLASHD1H.
The FLASHD1L and FLASHD1H are not used for programming the data flash memory. Data written to the FLASHD1L and FLASHD1H is invalid for programming the data falsh memory.

Description of bits

- **FD23 to FD016** (bits 7-0)
  The FD23 to FD16 bits are used to set the third byte of data.

- **FD31 to FD24** (bits 15-8)
  The FD31 to FD24 bits are used to set the fourth byte of data.

[Note]
- Specify a segment address to the FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.
- Erase data in the addresses in advance. Programmed data without erase cannot be guranteed.
- Do not access to unused area, otherwise the CPU may work incorrectly.

25.2.6 Flash Control Register (FLASHCON)

Address: 0xF096
Access: W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FLASHCON							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSERS	FERS
Access type	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

FLASHCON is a rewrite-only special function register (SFR) used to control the block erase and sector erase for the flash memory rewrite. FLASHCON returns 0x00 for reading.

Description of bits

- **FSERS** (bit 1), **FERS** (bit 0)
  FSERS bit and FERS bit are used to start the sector erase or block erase.
  Setting the FSERS bit or FERS bit to "1" starts erasing the sector or block specified by the FLASHSEG and FLASHAH register.

FSERS	FERS	Description
0	0	No function (initial value)
0	1	Start block erasing
1	0	Start sector erasing
1	1	Do not use (No function)

25.2.7 Flash Acceptor (FLASHACP)

Address: 0xF098
Access: W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FLASHACP							
Bit symbol	-	-	-	-	-	-	-	-	fac7	fac6	fac5	fac4	fac3	fac2	fac1	fac0
Access type	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

FLASHACP is a rewrite-only special function register (SFR) used to enable/disable the rewrite operation for flash memory.

Description of bits

- **fac7 to fac0** (bits 7-0)
  The fac7 to fac0 registers are used to restrict the erasing/rewrite operation in order to prevent an unintended erasing/rewrite operation.
  When “0FAH” and “0F5H” are written to FLASHACP in this order, the erasing or rewrite function is enabled only once. For subsequent erasing or rewrite, “0FAH” and “0F5H” must be written to FLASHACP each time. Even if another instruction is inserted between “0FAH” and “0F5H” written to FLASHACP, the erasing or rewrite function is enabled. Note that, if data other than “0F5H” is written to FLASHACP after “0FAH” is written, “0FAH” becomes invalid. In this case, write again from “0FAH”.

25.2.8 Flash Self Register (FLASHSLF)

Address: 0xF09C
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FLASHSLF							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSELF
Access type	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

FLASHSLF is a special function register (SFR) used to enable erasing and programming the flash memory.

Description of bits

- **FSELF** (bit 0)
  Set the FSELF bit to "1" to enable erasing and programming the flash memory.

FSELF	Description
0	Erasing and programming the flash memory is disabled (Initial value)
1	Erasing and programming the flash memory is enabled

[Note]
- Reset the FSELF bit to "0" before reading the Flash memory.

25.2.9 Flash Status Register (FLASHSTA)

Address: 0xF09E
Access: R
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								FLASHSTA							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FDPRSTA	FDERSTA
Access type	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

FLASHSTA is a special function register (SFR) used to check the flash memory status.
FLASHSTA is used when erasing or programming the data flash memory.
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 is completed.

Description of bits

- **FDERSTA** (bit 0)
  FDERSTA bit indicates if the data flash area is in the state of erasing.

FDERSTA	Description
0	Data flash memory is not in the state of erasing (Initial value)
1	Data flash memory is in the state of erasing

- **FDPRSTA** (bit 1)
  FDPRSTA bit indicates if the data flash area is in the state of programming.

FDPRSTA	Description
0	Data flash memory is not in the state of programming (Initial value)
1	Data flash memory is in the state of programming

[Note]
Perform the erasing or programming after checking the FDERSTA bit or FDPRSTA bit is “0”. The erasing or programming becomes invalid when the FDERSTA bit or FDPRSTA bit is “1”.

25.3 Self-programming

The self-programming function supports programming the program flash memory and data flash memory area by using specific function registers(SFRs).

- Programming the program flash memory area (segment n, n=0~)
  The block erase by 16Kbyte unit, the sector erase by 1Kbyte unit and reprogramming by 4Kbyte are available in the program flash memory area.
  The user program can be updated by using the remap function. See Chapter 2 “CPU and Memory Space” for detail about the remap function. For updating the user program, an another program must be pre-programmmed in a certain program code area untargetted by the erasing and self-programming.
  The CPU stops running the program codes during the block erase, sector erase and programming, and restarts running the next program code after the erase and programming have been completed.

- Programming the data flash memory area (segment 31)
  The block erase by 2Kbyte unit, the sector erase by 128byte unit and reprogramming by 1Kbyte are available in the data flash memory area. The CPU continues running the program code during the the block erase, sector erase and programming by the Back Ground Operation(BGO) function. Confirm the Flash Status register (FLASHSTA) to see if the block erase, sector erase or programming has been completed.

All bits of the target area in the flash memory get to “1” by the block erase or sector erase, and by programming data “0” turns the bits from “1” to “0”. When programming the flash memory, please erase the target area once. The data reprogrammed without erasing cannot be guranteed.
Use flash self register (FLASHSLF) and flash acceptor (FLASHACP) to prevent miss-erassing or miss-programming the flash memory. After enabling the programming function by the FLASHSLF and writing “0FAH” and “0F5H”(in order) to the FLASHACP, the block erase, sector erase and the programming gets enabled only once.
The erase and programming of flash memory is available when the high-speed clock is selected for the system clock. For details on how to switch the system clock, see Chapter 6 “Clock Generation Circuit”.

25.3.1 Notes on debugs for self-programming codes

Debug the program codes for self-programming on the U16 development environment(DTU8 debugger) in reference of the notes described in the Table 25-3.

Table 25-3 Notes on debugs for self-programming codes

Restricted function	Notes
Breakpoint break	Do not perform real time execution (GO execution) when the breakpoints are set in the flash self-programming sequence (from writing to the flash acceptor to writing to the flash data register). If the real time execution is performed with break points set in the sequence, the flash memory may not be reprogrammed.
Step execution	Do not perform STEP execution in the flash self-programming sequence (from writing to the flash acceptor to writing to the flash data register). If the STEP execution is performed in the sequence, the flash self-rewrite may not be reprogrammed.
Forced break	The DTU8 may respond an error message “ICE error: 6601H Failed to control target LSI. Please check the connection.” and after that, may not respond to any further debug commands. In that case, reset the DTU8, EASE1000 and the target system.

25.3.2 Programming the program memory

The block erase, sector erase and reprogramming operations are available for the program memory space. The units of block erase, sector erase and programming are 16 Kbytes, 1 Kbyte and 4 bytes, respectively. Figure 25-1 shows the flow chart for erasing the program memory.

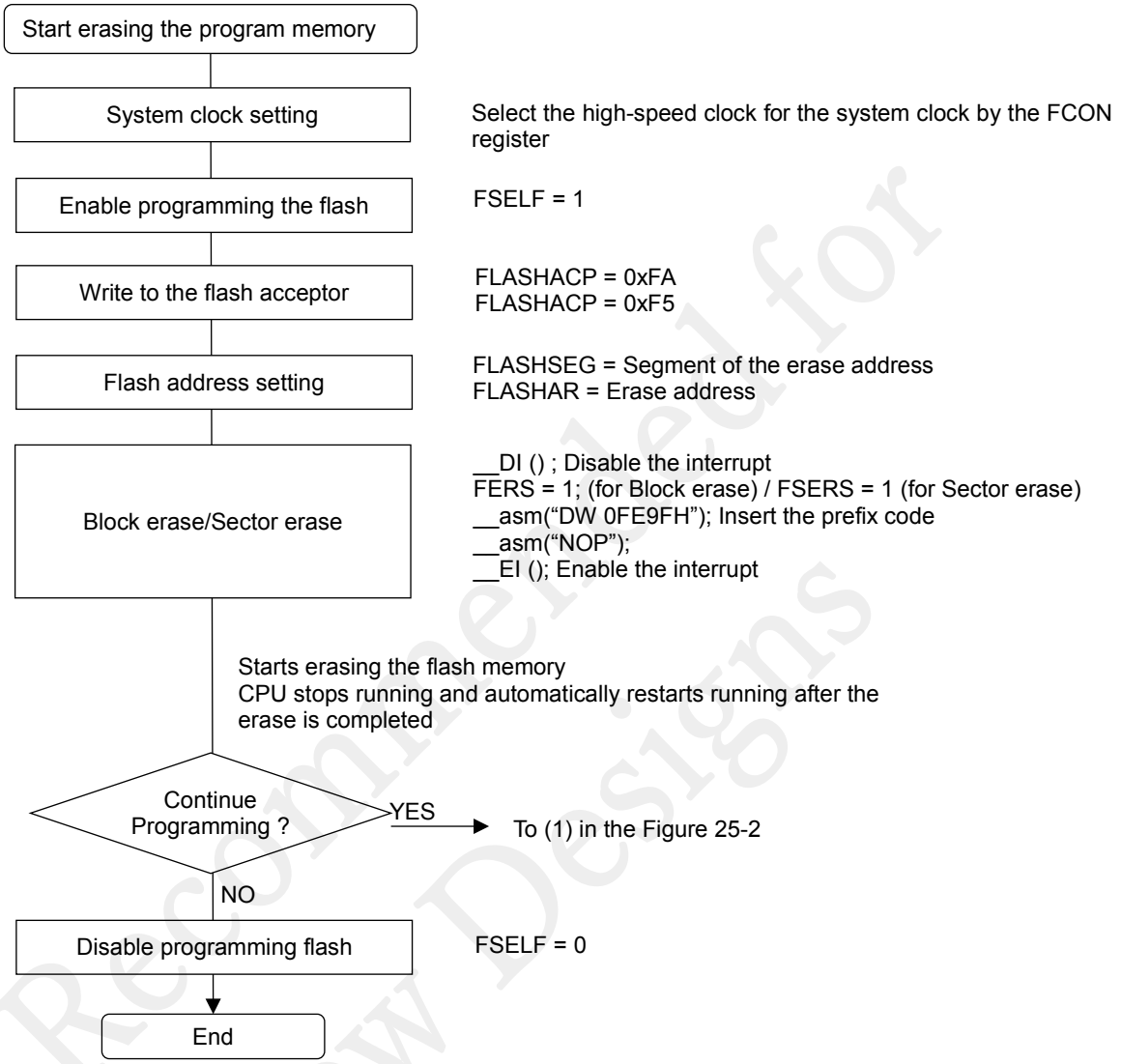


Figure 25-1 Flow chart for erasing the program memory

[Note]
- If any areas used for running the program were erased, the MCU would be uncontrollable by the software. Erase the areas unused for running the program.
- The CPU is in the wait state for max.85ms during the block erase or sector erase. Make a proper WDT clear process, so that it does get overflow during the erase.
- Interrupts are prohibited during the block erase or sector erase. Disable the interrupt by using __DI() function before setting FERS bit or FSERS bit.
- Insert the following codes on the next code of setting the FERS bit or FSERS bit, otherwise the operation is not guranteed.
  __asm("DW 0FE9FH");
  __asm("NOP");
- For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

Figure 25-2 shows the flow chart for programming the program memory.

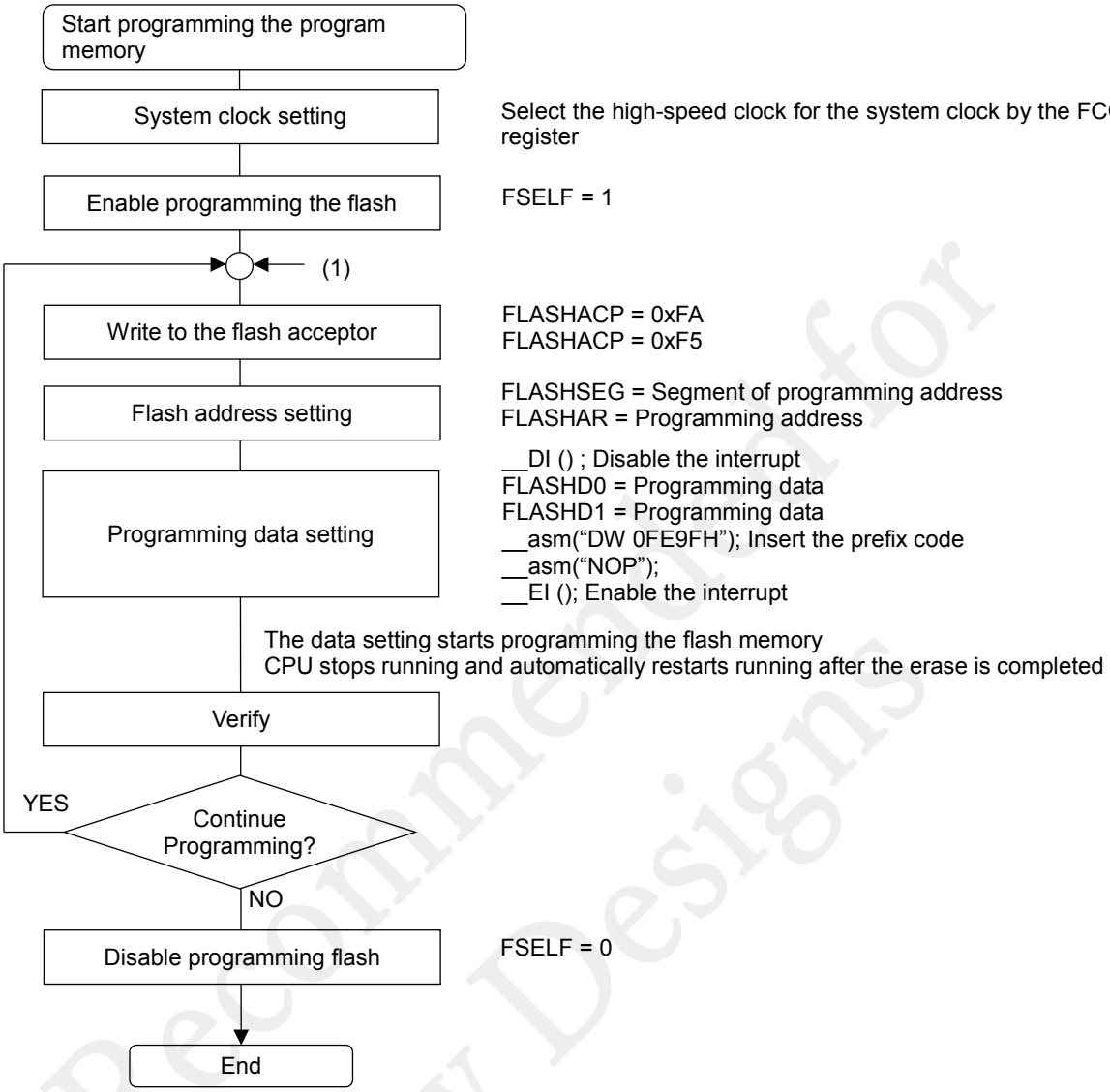


Figure 25-2 Flow chart for programming the program memory

[Note]
- If any areas used for running the program were reprogrammed, the MCU would be uncontrollable by the software. Program the areas unused for running the program.
- The CPU is in the wait state for max.80 μs during the block erase or sector erase. Make a proper WDT clear process, so that it does get overflow during the erase.
- Interrupts are prohibited during data flash write. Disable the interrupt by using __DI() function before setting write data to FLASHD1 register. Also, clear the WDT counter so that the WDT interrupt does not occur during the data set.
- Insert the following codes on the next code of setting the FLASHD1 register, otherwise the operation is not guranteed.

```
__asm("DW 0FE9FH");
__asm("NOP");
```

- For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

25.3.3 Programming the data flash memory

The block erase, sector erase and rewrite operations are available for the data flash memory area.
The block erase unit is 2 Kbytes, the sector erase unit is 128 bytes, and the rewrite unit is 1 byte.
The CPU continues running the program code during the the block erase, sector erase and programming by the Back Ground Operation(BGO) function.

Figure 25-3 shows the flow chart for erasing the data flash memory.

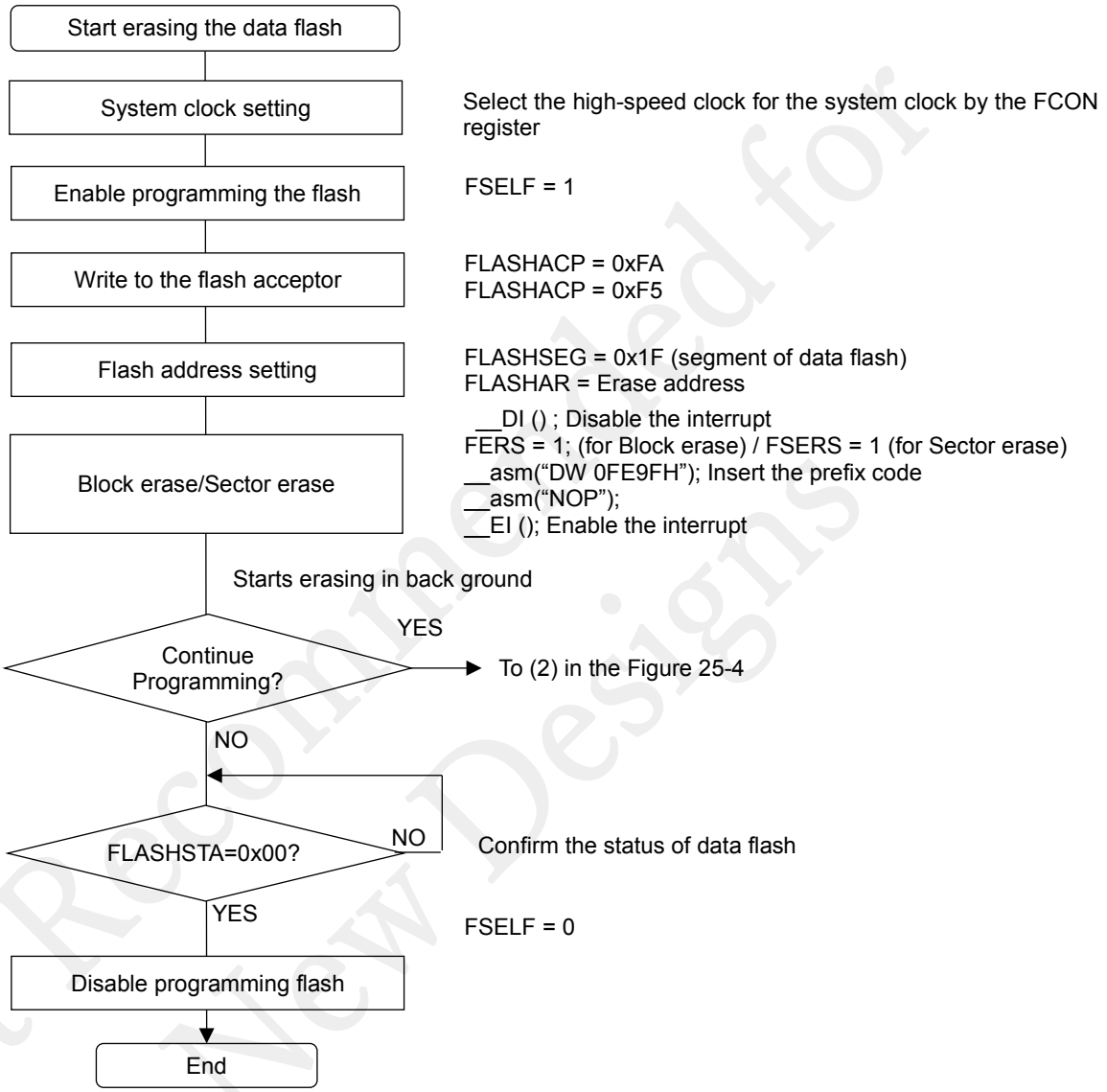


Figure 25-3 Flow chart for erasing the data flash memory

[Note]
- The CPU contines running the program code while erasing the data flash. Do not enter to STOP mode, STOP-D mode or HALT-H mode during the erase. Also, reset FSELF to "0" after the erase ended.
- The data flash areas are not readable during the erase.
- Interrupts are prohibited during the block erase or sector erase. Disable the interrupt by using __DI() function before setting FERS bit or FSERS bit.
- Insert the following codes on the next code of setting the FERS bit or FSERS bit, otherwise the operation is not guranteed.
  __asm("DW 0FE9FH");
  __asm("NOP");
- For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

Figure 25-4 shows the flow chart for programming the data flash memory.

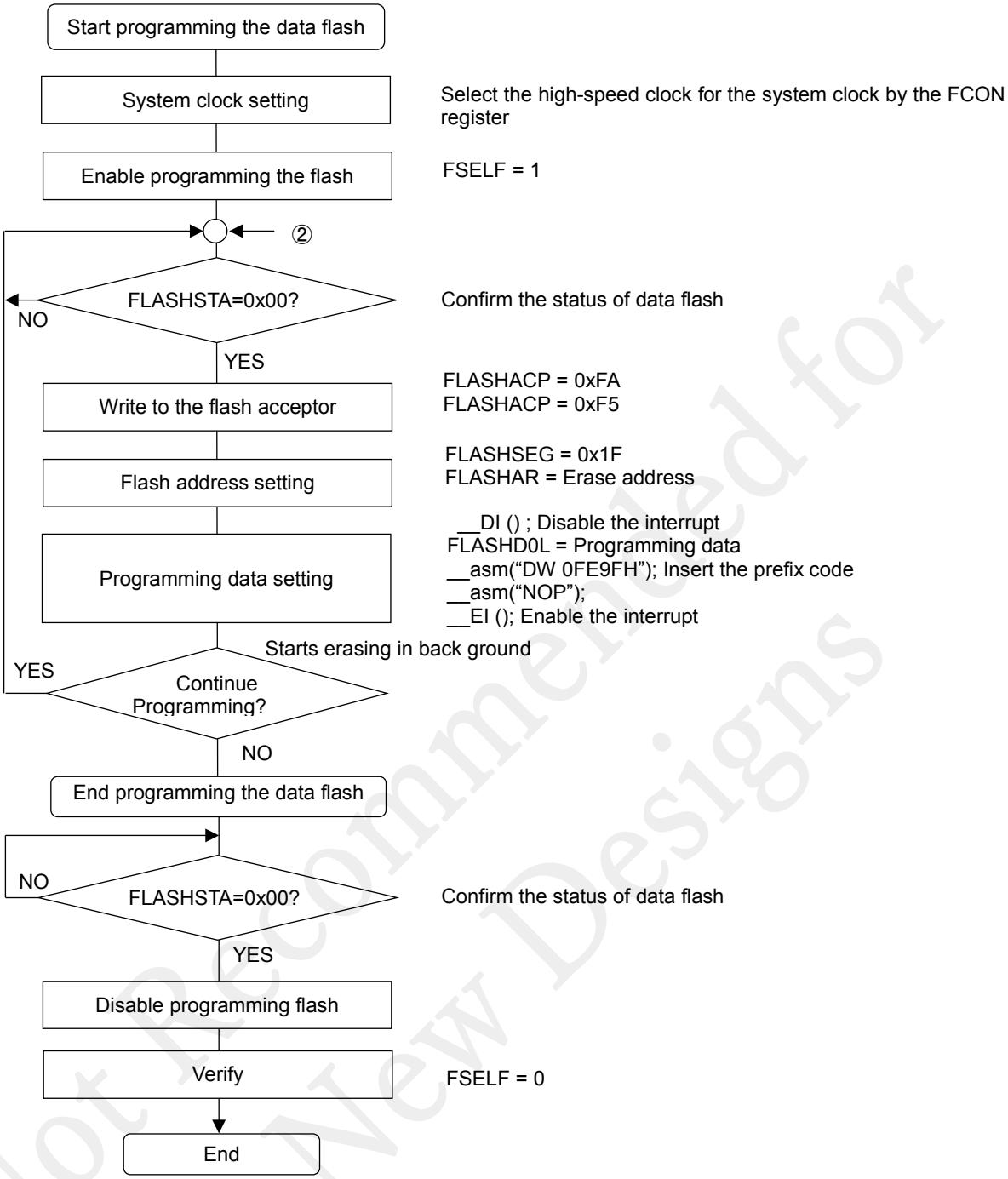


Figure 25-4 Flow chart for programming the data flash memory

[Note]
- The CPU contines running the program code while programming the data flash. Do not enter to STOP mode ,STOP-D mode or HALT-H mode while programming the data flash. Also, reset FSELF to "0" after the programming ended.
- The data flash areas are not readable while programming the data flash.
- Interrupts are prohibited during data flash write. Disable the interrupt by using __DI() function before setting write data to FLASHD0L register. Also, clear the WDT counter so that the WDT interrupt does not occur during the data set.
- Insert the following codes on the next code of setting the FLASHD0L register, otherwise the operation is not guranteed.
  __asm("DW 0FE9FH");
  __asm("NOP");
- For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

25.3.4 Notes in Use of the self-programming

- Enable and select the high-speed clock for the system clock when performing the block/sector erase or programming the flash memory. For details on how to enable the high-speed oscillation and switch the system clock, see Chapter 6 “Clock Generation Circuit”.
- In case a power failure occurred or the operation was terminated forcibly during block/sector erase or programming the flash memory, the programmed data cannot be guranteed. Please retry erasing and programming the flash memory.
- In case the power failure occurred or the operation was terminated forcibly during block/sector erase or programming the program memory area contains the address “0:0000H” and the MCU did not restart running, please reprogram a program code by using the on-chip debug emualtor (EASE1000) or using the ISP function.

25.4 ISP Function

ISP (In-System Programming) function allows users for programming the program flah memory and data flash memory by communicating to an external device.

25.4.1 Programming procedure

Figure 25-5 shows the flow chart of flash memory programming control.

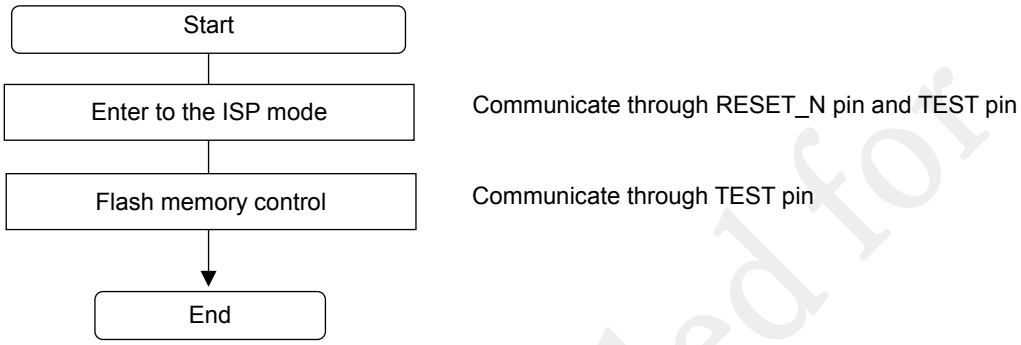


Figure 25-5 Flow Chart of Flash Memory programming control

25.4.2 Communication method

Both the RESET_N pin and TEST pin are used to enter the ISP mode and the TEST pin is used for the UART communication to control the command data. The UART works in a format of 8bit length, LSB first, one stop bit and no parity. The baudrate is available within a range min.4800bps to max.1Mbps.

25.4.3 Communication command

Three bytes commands are used to communicate with the MCU and control the ISP function , shown in the Table 25-4.

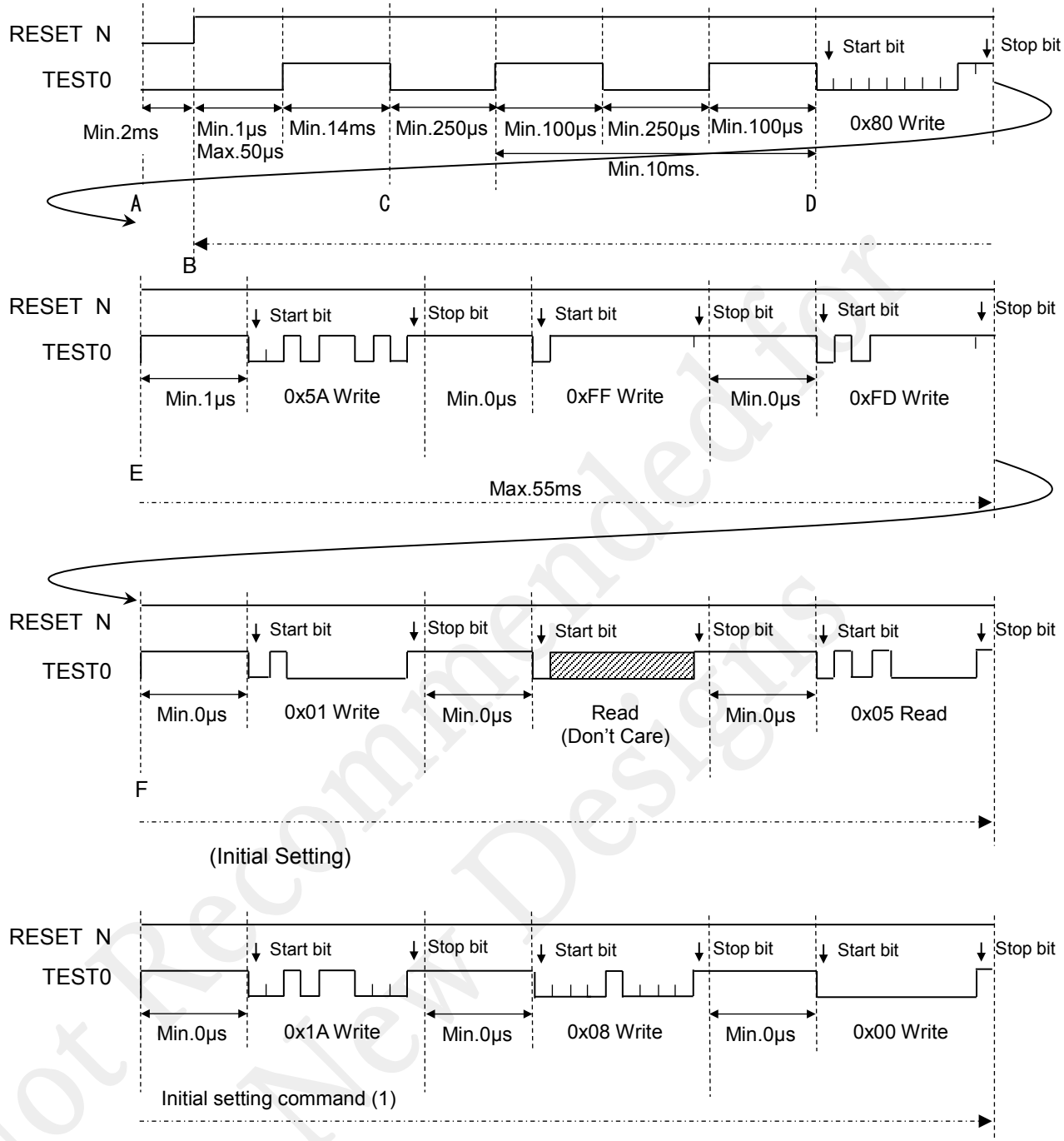
Table 25-4 ISP command list

Command		Fist byte	Second byte	Third byte
Initial setting	Initial setting command transmission (1)	0x1A	0x08	0x00
	Initial setting command transmission (2)	0x1A	0x00	0x00
	Initial setting command transmission (3)	0xC0	0x01	0x00
	Initial setting command transmission (4)	0xC0	0x05	0x00
	Initial setting command transmission (5)	0xC0	0x03	0x00
	Initial setting command transmission (6)	0xCE	0x01	0x00
	Initial setting command transmission (7)	0xCE	0x00	0x00
	Initial setting command transmission (8)	0x96	0xFF	0xFF
	Initial setting command transmission (9)	0x98	0xFF	0xFF
	Initial setting command transmission (10)	0x9A	0xFF	0xFF
	Initial setting command transmission (11)	0x9C	0xFF	0xFF
	Initial setting command transmission (12)	0x9E	0xFF	0xFF
	Command transmission completion confirmation (1)	0x01	(read) 0xC0 or 0x80	(read) 0x05
	Command transmission completion confirmation (2)	0x91	(read) 0x00	(read) 0x00
Common setting	Segment value setting	0xC6	0x00-0x1F (segment value)	0x00
	Address value setting	0xC8	Lower 8 bits	Higher 8 bits
	Busy signal confirmation	0xC5	0x1F (read)	0x01: BUSY 0x00: IDLE
Block erase	Block erase command	0xC2	0x05	0x00
Chip erase	Chip erase command	0xC2	0x06	0x00
For data program	Program Code area programming data (higher 2 bytes)	0xD2	Lower byte	Higher byte
	Program Code area programming data (lower 2 bytes)	0xCA	Lower byte	Higher byte
	Data Flash area programming data	0xCA	1-byte data	0xFF
	Programming command	0xC2	0x04	0x00
For verify	Program Code area expected data (higher 2 bytes)	0xE4	Lower byte	Higher byte
	Program Code area expected data (lower 2 bytes)	0xE2	Lower byte	Higher byte
	Data Flash area expected data	0xE4	1-byte data	0x00
	Verify command	0xC2	0x02	0x00
	Confirm collation result of expected values	0xE7	(read) 0x01: OK 0x00: NG	(read) 0x00

[Note]
- Programming 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.
- Programming the data flash area is performed in units of one byte.

25.4.4 ISP mode transition

The figure 25-6 shows the timing chart for entering the MCU to the ISP mode.



A. For the system reset, get the RESET_N pin to "L" level and TEST0 pin to "L" level.
B. Rise the RESET_N signal to "H" remaining the TEST signal "L".
C. Drive "L" level twice for min.250µs to the TEST pin.
D. Transmit data 0x80 to the TEST0 pin in the communication baud rate configured by the host side.
E. Transmit data to the TEST0 pin in the order of 0x5A→0xFF→0xFD in the communication baud rate configured by the host side.
F. Transmit data 0x01 to the TEST0 pin and repeat the procedure from A to F until the 3rd byte read data is 0x05 which indicates the communication baud rate matched.

Figure 25-6 Timing chart for entering the MCU to the ISP mode

[Note]
- Complete the ISP mode transfer command (point B in the Figure 25-6) through the end of the Initial setting command (1) within 55ms.

25.4.5 Flash Memory Control

Figure 25-7 shows the flow chart of flash memory erase and program, which are implemented after entering to the ISP mode.

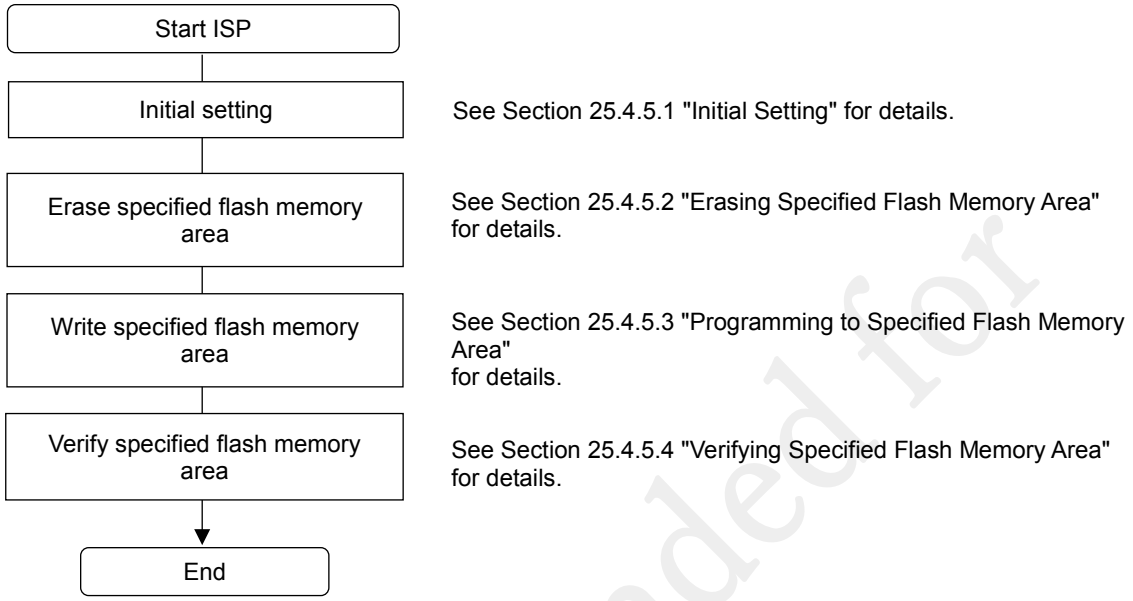


Figure 25-7 Flow chart of flash memory erase and program

25.4.5.1 Initialization

Figure 25-8 shows the flow chart of initialization.
The flash memory erase/program protection is released at the time of initial setting.

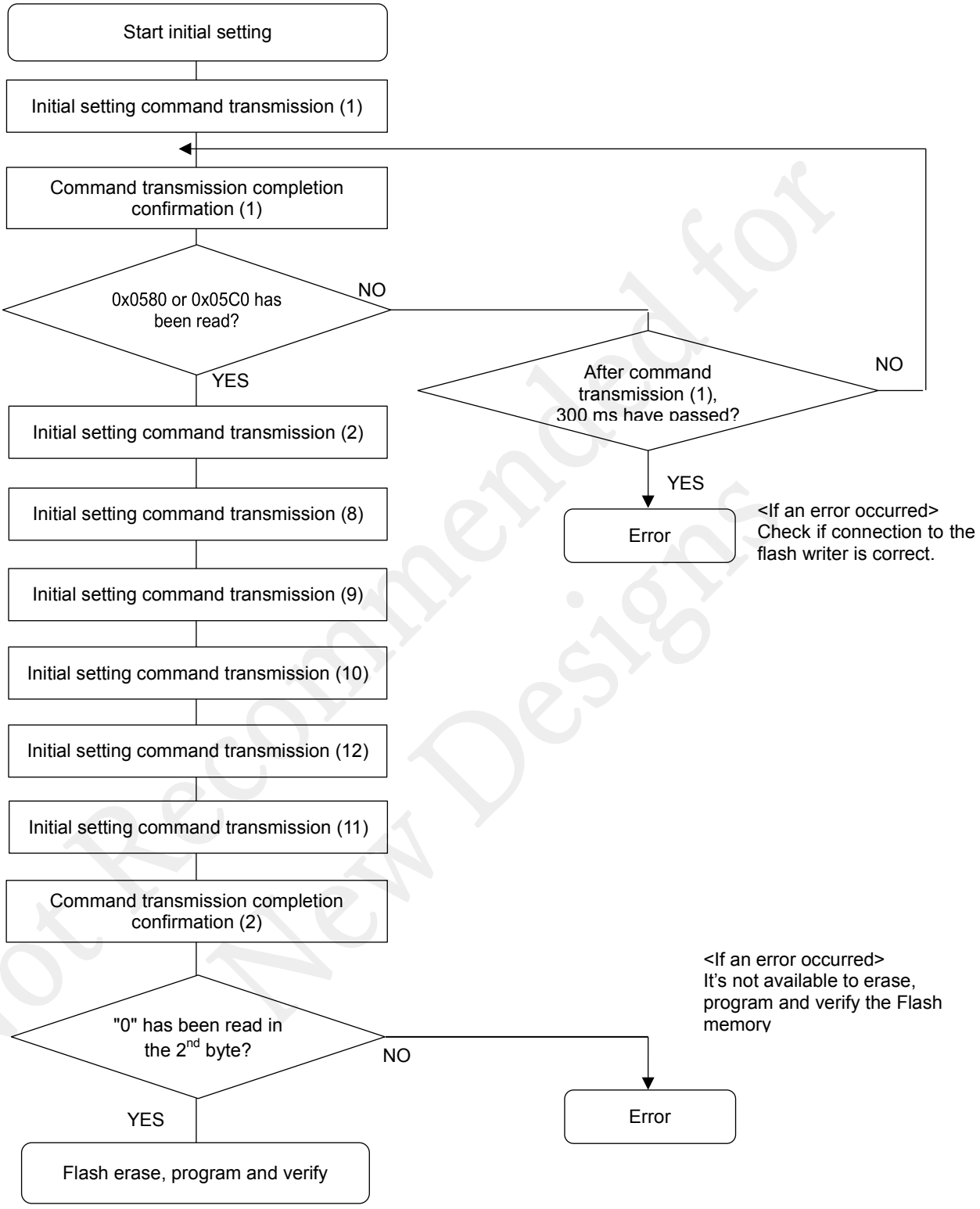


Figure 25-8 Initialization flow chart

25.4.5.2 Erase

Figure 25-9 shows the flow diagram for erasing the specified flash memory area.

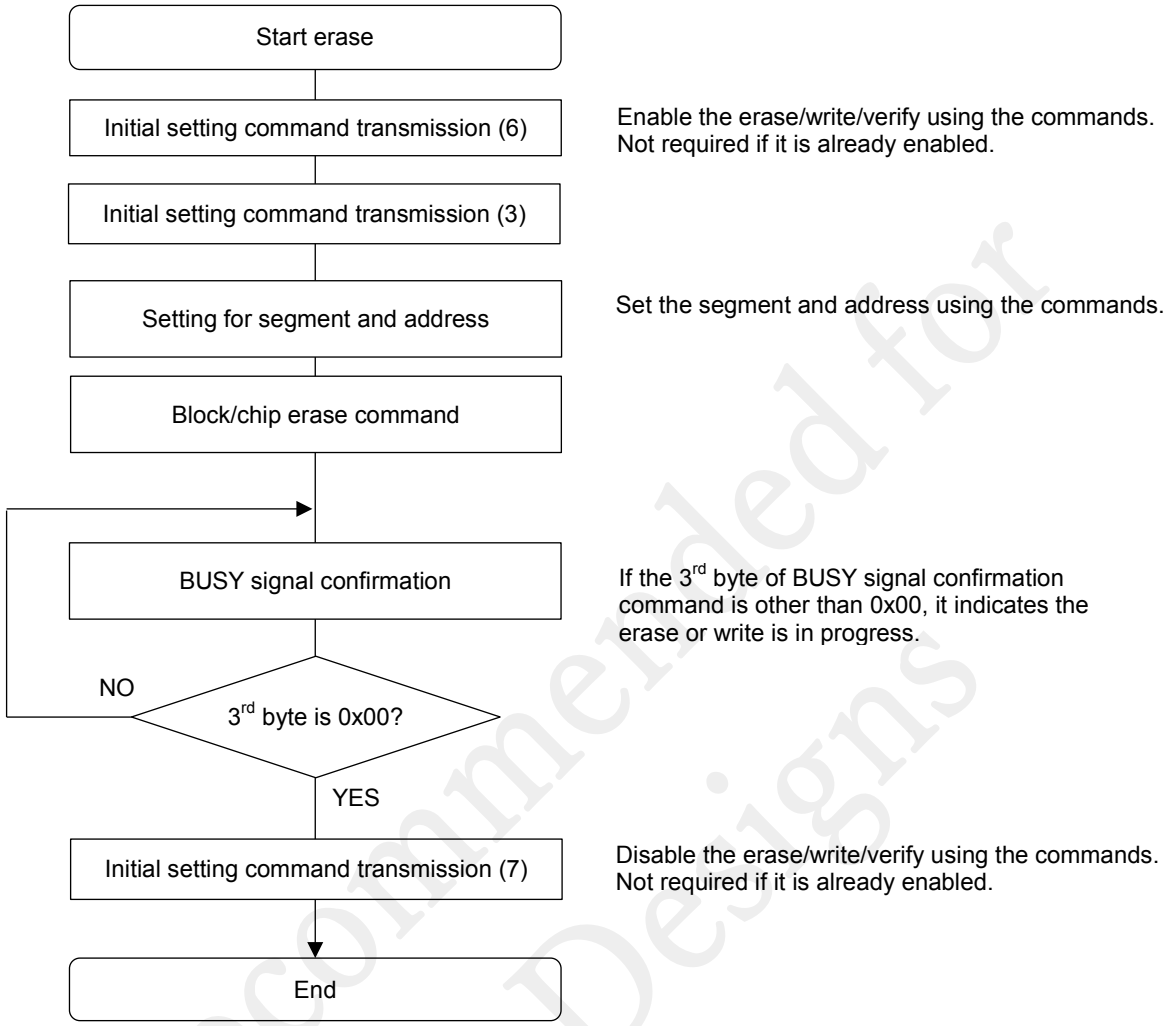


Figure 25-9 Flow Diagram for Erasing Specified Flash Memory Area

[Note]
- The erase process needs to be completed within 500ms.

25.4.5.3 Programming

Figure 25-10 shows the flow diagram for programming to the specified flash memory area.

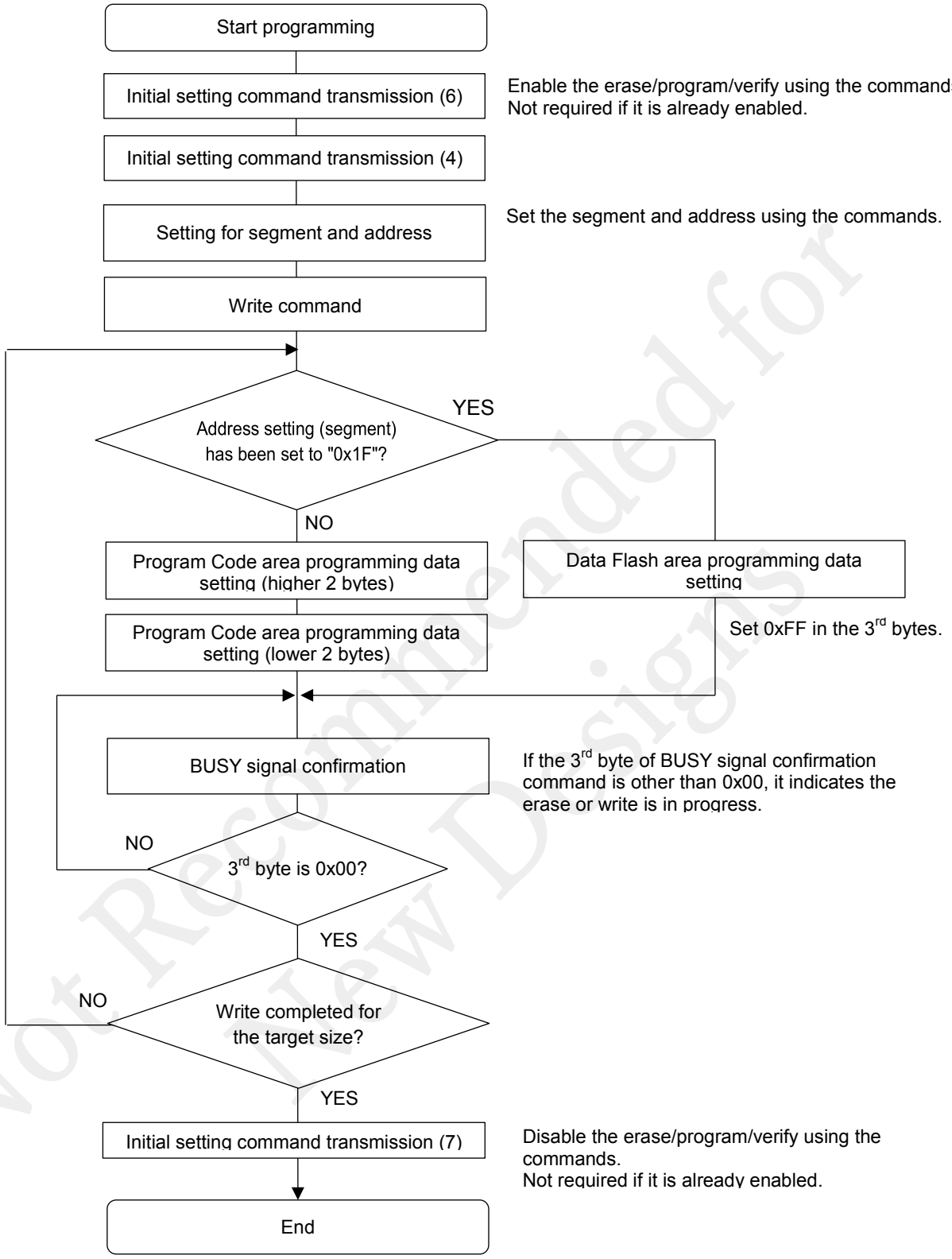


Figure 25-10 Flow Diagram for Programming to Specified Flash Memory Area

[Note]
- The programming process needs to be completed within 500ms. In the case of programming to multiple addresses, the process from previous setting data to the next setting data or to the end of initial setting command transmission (7) within 500ms.

25.4.5.4 Verification

Figure 25-11 shows the flow diagram for verifying the specified flash memory area.

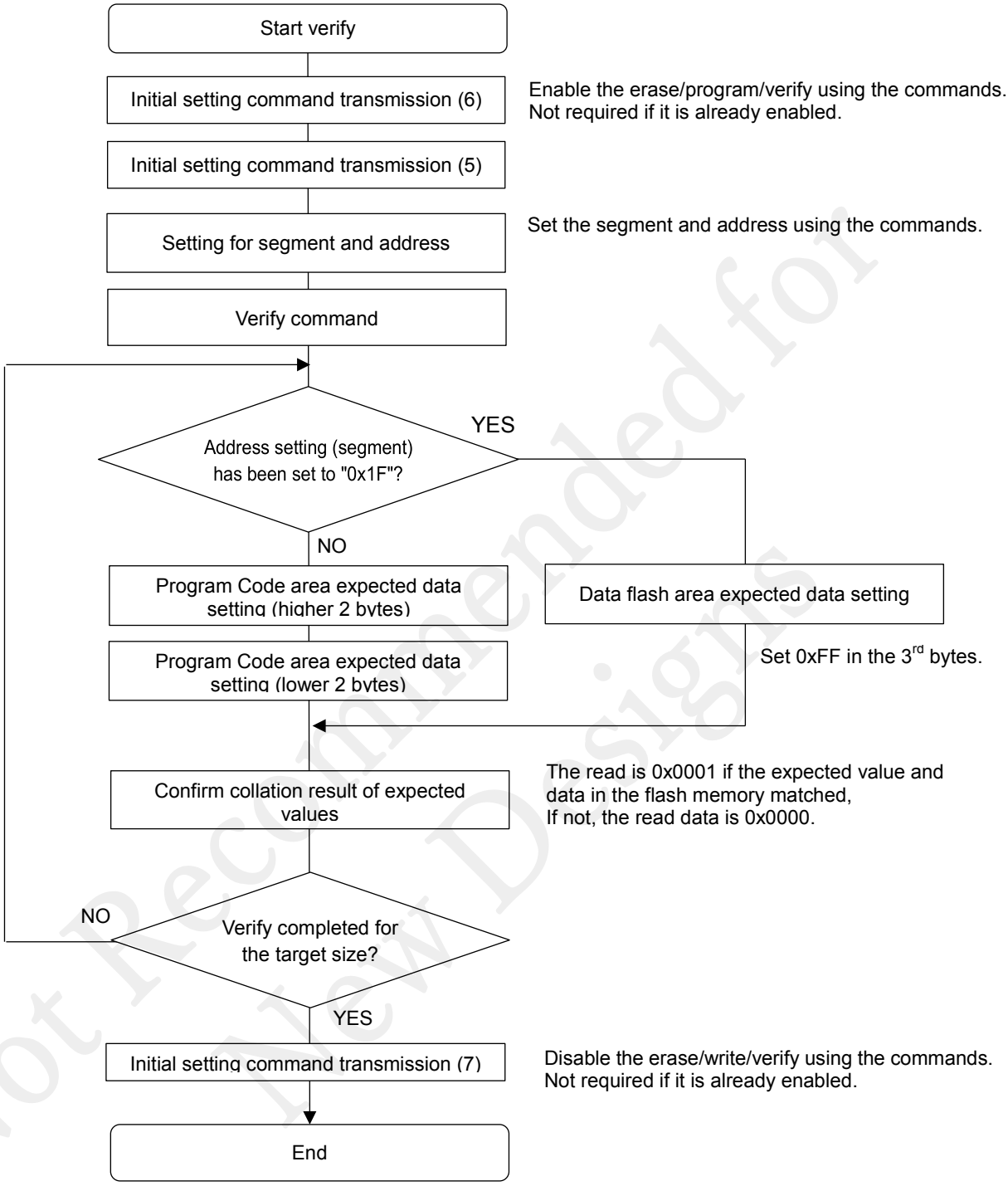


Figure 25-11 Flow Diagram for Verifying Specified Flash Memory Area

[Note]
- The verify process needs to be completed within 500ms. In the case of verifying multiple addresses, the process from previous setting data to the next setting data or to the end of initial setting command transmission (7) within 500ms.

Chapter 26 Code Option

26. Code Option

26.1 General Description

ML62Q1000 series has the code option.
The hardware refers to the setting of code option at the power-up or system reset to activate the functions.
The code option provides the functions to select the CPU operating mode, the PLL reference frequency, WDT operation mode, enable/disable the unused ROM area access reset and the remap function. The code option area is in the address of program memory.
The code option area is programmable by using the on-chip debug function, self-programming function and ISP function.
For details abou the on-chip debug function, self-programming function and ISP function, see Chapter 28 “On-chip debug function” and Chapter 25 “Flash Memory”.
For details of the configuration of program memory, see Chapter 2 “CPU and Memory Space”.

26.1.1 Features

- Selection of CPU operating mode (Wait mode and No wait mode)
- Selection of PLL reference frequency (16MHz, 24MHz and 32 MHz)
- Selection of WDT operation mode (enable/disable and operation clock)
- Enable/disable the unused ROM area reset function
- Enable/disable the remap function

26.2 Description of Code Option

26.2.1 Code Options 0 (CODEOP0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CODEOP0															
Bit symbol	-	-	-	PCERMD	-	-	-	REMAPMD	-	-	-	-	-	WDTNMCK	WDTSPMD	WDTMD

Description of bits

- **WDTMD** (bit 0)
  The WDTMD bit is used to enable/disable WDT operation.

WDTMD	Description
0	Disable the WDT operation
1	Enable the WDT operation (Initial value)

- **WDTSPMD** (bit 1)
  Set always the WDTSPMD bit to “0”.

- **WDTNMCK** (bit 2)
  The WDTNMCK bit is used to select the WDT operation clock.

WDTNMCK	Description
0	WDT operates with approx.1.024kHz divided from the low-speed RC oscillation clock (approx. 32.768 kHz)
1	RC1K clock (RC oscillation 1KHz clock for WDT)

- **REMAPMD** (bit 8)
  The REMAPMD bit is used to enable/disable the software remap function.

REMAPMD	Description
0	Enable the remap function
1	Disable the remap function (Initial value)

- **PCERMD** (bit 12)
  The PCERMD bit is used to enable/disable the unused ROM are access resset in safery function that makes the reset in case the PC (Program Counter) indicated non-existing address of ROM area.

PCERMD	Description
0	Disable the invalid memory access reset
1	Enable the invalid memory access reset (Initial value)

26.2.2 Code Options 1 (CODEOP1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	CODEOP1															
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	PLLMD1	PLLMD0	CPUMD1	CPUMD0

Description of bits

- **CPUMD1 to 0** (bits 1-0)
  The CPCUMD1 to 0 bits are used to select the operation mode of CPU.
  See Chapter 2 “CPU and Memory Space” for details about the operation mode (wait mode and no wait mode).

CPUMD1	CPUMD0	Description
0	0	Prohibited to use (Wait mode)
0	1	Wait mode
1	0	Prohibited to use (No wait mode)
1	1	No wait mode (Initial value)

- **PLLMD1 to 0** (bits 3-2)
  The PLLMD1 to 0 bits are used to select the PLL reference frequency.

PLLMD1	PLLMD0	Description
0	0	Prohibited to use (PLL reference frequency = 32 MHz)
0	1	PLL reference frequency = 32 MHz
1	0	PLL reference frequency = 24 MHz
1	1	PLL reference frequency = 16 MHz (initial value)

The maximum operating frequency is limitted in each case of PLL frequency as follows.

PLL oscillation mode	Maximum operating frequency		
	Peripheral	CPU (Wait mode)	CPU (No wait mode)
32MHz mode	32MHz	16MHz	8MHz
24MHz mode	24MHz	24MHz	12MHz
16MHz mode	16MHz	16MHz	8MHz

26.3 How to configure the Code Option data

The address of code option area is defferent product by product, depends on the size of program flash memory.

Table 26-1 Code option address list

Product	Program Flash Memory Size	Code Option Area	Address	
			CODEOP1	CODEOP0
ML62Q1223A/1233A	16K byte	0:3FC0 - 0:3FDF	0:3FD2	0:3FD0
ML62Q1224A/1234A	24K byte	0:5FC0 - 0:5FDF	0:5FD2	0:5FD0
ML62Q1225A/1235A/1245A/1265A	32K byte	0:7FC0 - 0:7FDF	0:7FD2	0:7FD0
ML62Q1246A/1266A	48K byte	0:0BFC0 - 0:0BFDF	0:0BFD2	0:0BFD0
ML62Q1247A/1267A	64K byte	0:0FFC0 - 0:0FFDF	0:0FFD2	0:0FFD0

Figure 26-1 shows an example of defining the code option data(in the case of the product that has 64Kbyte program memory).
The start-up file for each product (ML62xxxx.ASM) includes these code.

```
;-------------------------------------------------------
;       Setting the code-option data
;-------------------------------------------------------
cseg at     0ffc0h          ;address
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0eef8h           ; CODEOP0
                             ;WDT is disabled, remap function is enabled and unused ROM area access reset is disabled)
      dw    0fff9h           ; CODEOP1
                             ; PLL =24MHz and CPU wait mode
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
      dw    0ffffh           ; Fixed (FFFFH)
```

Figure 26-1 Code Option Data Definition Example (Program Memory Space = 64 Kbyte Product)

[Note]
- Be sure to set the addresses or bits not used in the code option data area to “1”.
- The code option area of the blank part is filled with 0xFFFF in the factory setting.

Chapter 27 LCD Driver

ML62Q1200A group does not have this function which is built-in other products of ML62Q1000 series

Chapter 28 On-Chip Debug Function

28. On chip debug function

28.1 General Description

ML62Q1000 series has the on-chip debug function, which is used to connect the MCU and the host PC through the on-chip emulator (EASE1000).
On-board debugging and programming the program code are available by using using DTU8 debugger.

28.2 Connection with the On-chip Debug Emulator EASE1000

Figure28-1 shows the connection example of when to use the power from the EASE100(+3.3V/100mA).

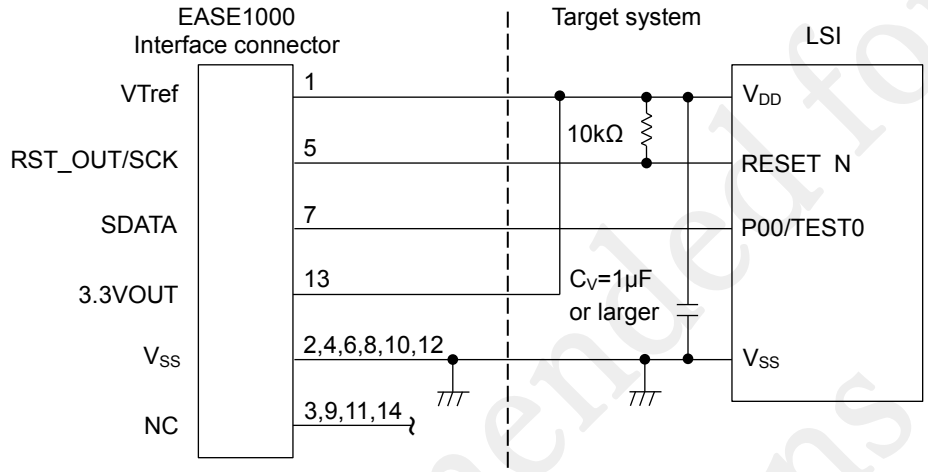


Figure 28-1 EASE1000 Connection Example when to use the power from EASE1000

Figure28-2 shows the connection example when to use the power from the target system (+1.6V to 5.5V).

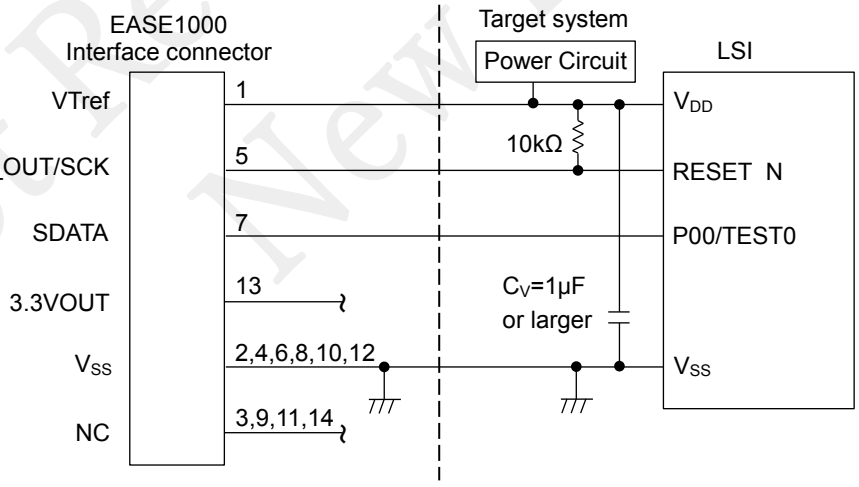


Figure 28-2 EASE1000 Connection Example when to use the power from the target system

28.3 Notes on Debug with EASE1000

- Do not mount a component that makes the level fixed to H on the RESET_N pin.
- Do not mount a component on the P00/TEST0 pin.
- Do not write an application code that sets P00 to the output mode to the LSI. Since this code is executed before EASE1000 accesses the LSI, the P00/TEST0 pin is set to the output mode, so the LSI can no longer connect the EASE1000. Please note that the input/output mode of P00 cannot be initialized from EASE1000.
- Validate the ROM code on user production board without LAPIS semiconductor development tool EASE1000.
- Do not connect EASE1000 when measure the consumption current of a target system. The on-chip debugging circuit in target LSI influences, and consumption current increases.
- When using the 3.3VOUT power supply of EASE1000, do not apply power of the user target system to the VDD pin of the target LSI. If both power supplies are connected, it may damage EASE1000 or cause an electric shock or fire.
- Supply 3.0V to 5.5V to the V_{DD} pin when programming the Flash on the MCU by using the EASE1000.
- Please do not apply LSIs being used for debugging to mass production.

28.4 Overview of On-Chip Debug Function

This section shows debug functions available by connecting EASE1000 with the LSI and using the DTU8 debugger. See "DTU8 user's manual" for more details about the DTU8 debugger.

- Emulation
  - Real time emulation
  - Single step emulation
- Break
  - Hardware break point break (7 points)
  - RAM data matching break
  - Sequential break
  - Trace overflow break
  - Stack overflow/underflow break
  - ROM unused area access break
  - RAM parity error break
- Trace
  - Branch trace
- Real time watch
- CPU resource display/change
  - Internal ROM reference/disassembly
  - Internal RAM/SFR display/change
  - CPU internal register display/change
- Application Program download
  - Application program download or erase to internal flash Memory
  - Application program read from internal flash Memory
  - Data write or erase to internal data flash Memory
  - Data read from internal data flash Memory
- Peripheral operation continuation/stop control during break

[Note]
- The ROM unused area access reset does not occur in the on-chip debug mode. The ROM unused area access break can be generated by setting the function on the DTU8 debugger.
- The RAM parity error reset does not occur in the on-chip debug mode. The RAM parity error break can be generated on the DTU8 debugger.

28.4.1 Peripheral Operation Continuation/Stop Control during Break

DTU8 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 opration is controlled.

Table 28-1 Peripheral controls during the break on the DTU8

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. Check always the item on. There are some cases that the expanded external interrupt status register get cleared.
Low-speed Time Base Counter	Low-speed Time Base Counter and Buzzer	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. [Note] - The simplified RTC functions (LTBRR register and TBCOUT1 output) stop during the break even if this item is checked on. - The buzzer function is also controlled by this item, because the low-speed time base counter supplies T1HZ and T8HZ signals to the buzzer circuit.
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.
Functional Timer	Functional Timer	
Serial Unit (SIO/UART)	Serial Communication Unit	
I^2C Bus Unit (Master/Slave)	I^2C Bus Unit	
I^2C Master Module	I^2C Bus Master	
Buzzer	Buzzer	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. Checked on also the item "Low-speed Time Base Counter" when continuing the operation during the break.
Analog Module (CMP/ADC/VLS)	Successive Approximation type A/D Converter, Analog Comparator and Voltage Level Supervisor(VLS)	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.
DMA Controller	DMA Controller	

Chapter 29 Safety Function

29. Safety Function

29.1 General Description

ML62Q1000 series has the safety functions to make a safe stop in case a failure is detected by executing the self-diagnosis software, available to support IEC60730/60335.

29.1.1 Features

- RAM guard
  Protest the miss-writing to the RAM (controlled by SFR).
- SFR guard
  Protest the miss-writing to the SFR (controlled by SFR).
- Successive approximation type A/D converter test
  Successive approximation type AD converter test function (controlled by SFR)
- 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).
- 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 normal oscillation of the high-speed and low-speed clocks (controlled by SFR).
- CRC（Cyclic Redundancy Check）calculation
  Detect data errors in flash memory (controlled by SFR)
- UART test
  UART self test function (controlled by SFR).
- SSIO test
  SSIO self test function (controlled by SFR).
- I^2C test
  I^2C self test function (controlled by SFR).
- WDT counter read
  Detect WDT operation by reading the WDT counter value (controlled by SFR).
- GPIO test
  GPIO self test function (controlled by SFR)

29.2 Description of Registers

29.2.1 List of Registers

Address	Name	Symbol (byte)	Symbol (word)	R/W	Size	Initial value
0xF0B0	RAM guard setting register	RAMGD	-	R/W	8	0x00
0xF0B1	Reserved register	-	-	R	8	0x00
0xF0B2	Reserved register	-	-	R	8	0x00
0xF0B3	Reserved register	-	-	R	8	0x00
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	Reserved register	-	-	R	8	0x00
0xF0B9	Reserved register	-	-	R	8	0x00
0xF0BA	Successive approximation type A/D converter test function	ADSFMOD	-	R/W	8	0x00
0xF0BB	Reserved register	-	-	R	8	0x00
0xF0BC	RAM parity setting register	RASFMOD	-	R/W	8	0x00
0xF0BD	Reserved register	-	-	R	8	0x00
0xF0BE	Communication test setting register 0	COMFT0L	COMFT0	R/W	8/16	0x00
0xF0BF		COMFT0H		R/W	8	0x00

[Note]
Word access is available for the registers having word symbols. To perform a word access, specify an even address.

29.2.2 RAM Guard Setting Register (RAMGD)

Address: 0xF0B0
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								RAMGD							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	RGD2	RGD1	RGD0
Access type	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

RAMGD is a special function register (SFR) used to guard the RAM from the write access.
Specified RAM area is protectable.

Description of bits

- **RGD2 to RGD0** (bits 2-0)
  The RGD2 to RGD0 bits are used to select RAM area to protect.

RGD2	RGD1	RGD0	Description
0	0	0	All RAM area writable and readable (initial value)
0	0	1	0:0xEFC0 to 0:0xEFFF (64B) unwritable and readable
0	1	0	0:0xEF80 to 0:0xEFFF (128B) unwritable and readable
0	1	1	0:0xEF00 to 0:0xEFFF (256B) unwritable and readable
1	0	0	0:0xEE00 to 0:0xEFFF (512B) unwritable and readable
1	0	1	Setting prohibited (0:0xEE00 to 0:0xEFFF (512B) unwritable and readable)
1	1	0	Setting prohibited (0:0xEE00 to 0:0xEFFF (512B) unwritable and readable)
1	1	1	Setting prohibited (0:0xEE00 to 0:0xEFFF (512B) unwritable and readable)

29.2.3 SFR Guard Setting Register 0 (SFRGD0)

Address: 0xF0B4
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SFRGD0															
Byte symbol	SFRGD0H								SFRGD0L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	SGD05	SGD04	SGD03	SGD02	SGD01	SGD00
Access type	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

SFRGD0 is a special function register (SFR) used to gurad SFR from write access.
Specified SFR of the peripherals is protectable.

Description of setting value

Setting value	Description
0	SFR writable and readable (initial value)
1	SFR is unwritable and readable

Corresponding block

Bit	Bit symbol	Corresponding interrupt
Bit 15	SGD0F	-
Bit 14	SGD0E	-
Bit 13	SGD0D	-
Bit 12	SGD0C	-
Bit 11	SGD0B	-
Bit 10	SGD0A	-
Bit 9	SGD09	-
Bit 8	SGD08	-
Bit 7	SGD07	-
Bit 6	SGD06	-
Bit 5	SGD05	Watchdog timer mode register (WDTMOD) described in Chapter 10 “Watchdog Timer”
Bit 4	SGD04	Block control register 0 to 3 (BCKCON0 to 3) and block reset control register 0 to 3 (BRECON0 to 3) described in Chapter 4 “Power Management”
Bit 3	SGD03	RAM parity function setting register (RASFMOD) in this chapter
Bit 2	SGD02	SFR described in Chapter 22 “Voltage Level Supervisor (VLS0)”
Bit 1	SGD01	SFR described in Chapter 6 “Clock Generation Circuit”
Bit 0	SGD00	SFR described in Chapter 5 “Interrupt”

29.2.4 SFR Guard Setting Register 1 (SFRGD1)

Address: 0xF0B6
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	SFRGD1															
Byte symbol	SFRGD1H								SFRGD1L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	SGD13	SGD12	SGD11	SGD10
Access type	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

SFRGD1 is a special function register (SFR) used to gurad SFR from write access.
Specified SFR of the peripherals is protectable.

Description of setting value

Setting value	Description
0	SFR writable and readable (initial value)
1	SFR is unwritable and readable

Corresponding block

Bit	Bit symbol	Corresponding interrupt
Bit 15	SGD1F	-
Bit 14	SGD1E	-
Bit 13	SGD1D	-
Bit 12	SGD1C	-
Bit 11	SGD1B	-
Bit 10	SGD1A	-
Bit 9	SGD19	-
Bit 8	SGD18	-
Bit 7	SGD17	-
Bit 6	SGD16	-
Bit 5	SGD15	-
Bit 4	SGD14	-
Bit 3	SGD13	SFR related to GPIO3 described in Chapter 17 “GPIO”
Bit 2	SGD12	SFR related to GPIO2 described in Chapter 17 “GPIO”
Bit 1	SGD11	SFR related to GPIO1 described in Chapter 17 “GPIO”
Bit 0	SGD10	SFR related to GPIO0 described in Chapter 17 “GPIO”

29.2.5 RAM Parity Setting Register (RASFMOD)

Address: 0xF0BC
Access: R/W
Access size: 8 bits
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								RASFMOD							
Bit symbol	-	-	-	-	-	-	-	-	PERF	-	-	-	-	-	-	PEREN
Access type	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

RASFMOD is a special function register (SFR) used to control the RAM parity error reset function.
The RAM parity error detection and the RAM parity error reset are selectable.
The reset flag by a RAM parity error can be checked by the reset status register (SRSTAT).
For the flag, see Chapter 3 “Reset Function”.

Description of bits

- **PEREN** (bit 0)
  The PEREN bit is used to enable/disable the RAM parity error reset function.

PEREN	Description
0	RAM parity error reset function disabled (initial value)
1	RAM parity error reset function enabled

- **PERF** (bit 7)
  The PERF bit is used to check whether a RAM parity error occurs. Write “1” 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.

PERF	Description
0	No RAM parity error (initial value)
1	RAM parity error occurred

29.2.6 Communication Test Setting Register (COMFT0)

Address: 0xF0BE
Access: R/W
Access size: 8/16 bits
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	COMFT0															
Byte symbol	COMFT0H								COMFT0L							
Bit symbol	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMFT01	CMFT00
Access type	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

COMFT0 is a special function register (SFR) used to control the communication test function, which enables the loop back test with transmit data in the serial communication units.
As the I2C bus unit and I2C master are equipped with the function to read the transmit data, the function can be used for testing. For details, see Chapter 12 "I^2C Buss Unit" and Chapter 13 "I^2C Master".

Description of setting value

Setting value	Description
0	Communication function self-test disabled (initial value)
1	Communication function self-test enabled

Corresponding block

Bit	Bit symbol	Corresponding interrupt
Bit 15	CMFT0F	-
Bit 14	CMFT0E	-
Bit 13	CMFT0D	-
Bit 12	CMFT0C	-
Bit 11	CMFT0B	-
Bit 10	CMFT0A	-
Bit 9	CMFT09	-
Bit 8	CMFT08	-
Bit 7	CMFT07	-
Bit 6	CMFT06	-
Bit 5	CMFT05	-
Bit 4	CMFT04	-
Bit 3	CMFT03	-
Bit 2	CMFT02	-
Bit 1	CMFT01	Serial unit 1
Bit 0	CMFT00	Serial unit 0

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 and SSIO (synchronous serial port) of the serial communication unit.
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 (2nd to 8th function) of the general-purpose port.
For receive side data, it is not required to set the mode (2nd to 8th 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.

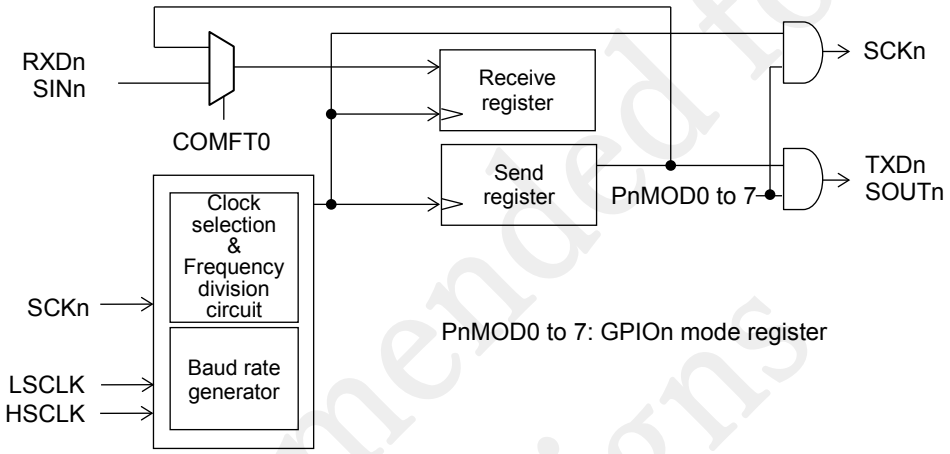


Figure 29-1 Communication Test Concept Diagram

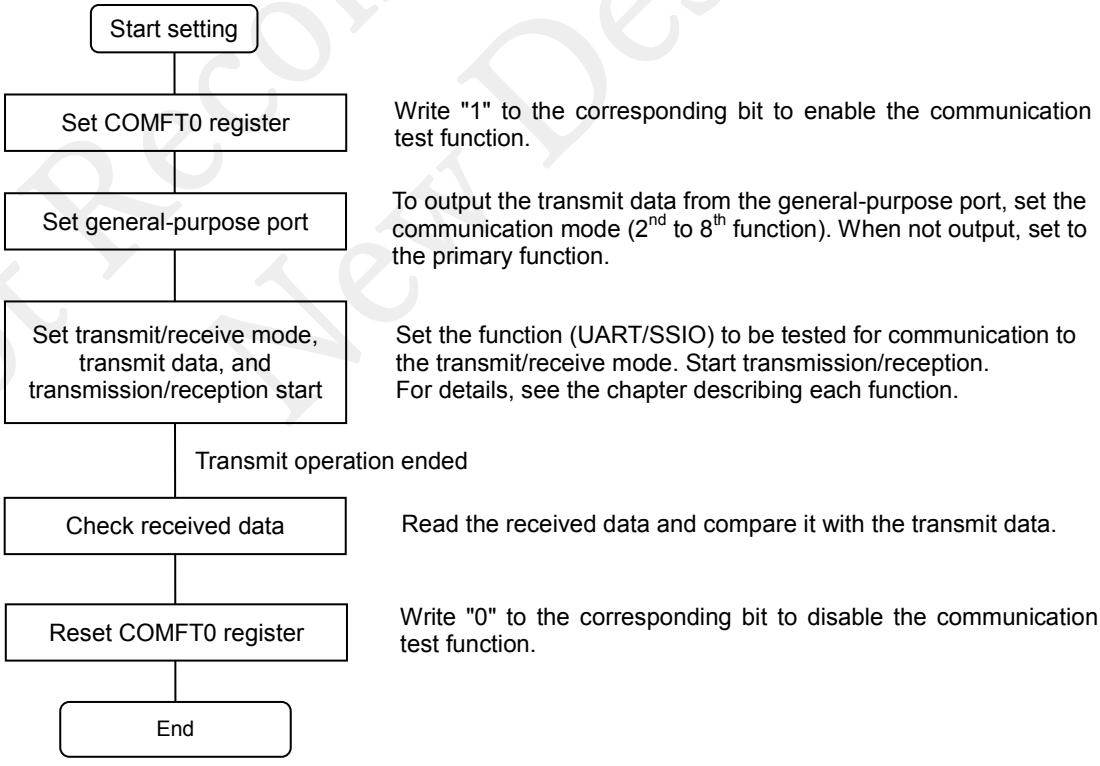


Figure 29-2 Communication Test Flow Chart

29.3.2 ROM Unused 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>

ML62Q1267A	:	0:0xFFC0 to 31:0xFFFF
ML62Q1266A	:	0:0xBFC0 to 31:0xFFFF
ML62Q1265A	:	0:0x7FC0 to 31:0xFFFF
ML62Q1247A	:	0:0xFFC0 to 31:0xFFFF
ML62Q1246A	:	0:0xBFC0 to 31:0xFFFF
ML62Q1245A	:	0:0x7FC0 to 31:0xFFFF
ML62Q1235A	:	0:0x7FC0 to 31:0xFFFF
ML62Q1234A	:	0:0x5FC0 to 31:0xFFFF
ML62Q1233A	:	0:0x3FC0 to 31:0xFFFF
ML62Q1225A	:	0:0x7FC0 to 31:0xFFFF
ML62Q1224A	:	0:0x5FC0 to 31:0xFFFF
ML62Q1223A	:	0:0x3FC0 to 31:0xFFFF

[Note]

- **CSR[3] is unused on the ML62Q1000 series. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".**

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 functional timer are available to implement the function.

LSCLK is countable by a trigger of the clock for mutual monitoring, enables to monitor mutually the two oscillation clocks.
See the application note for more details.

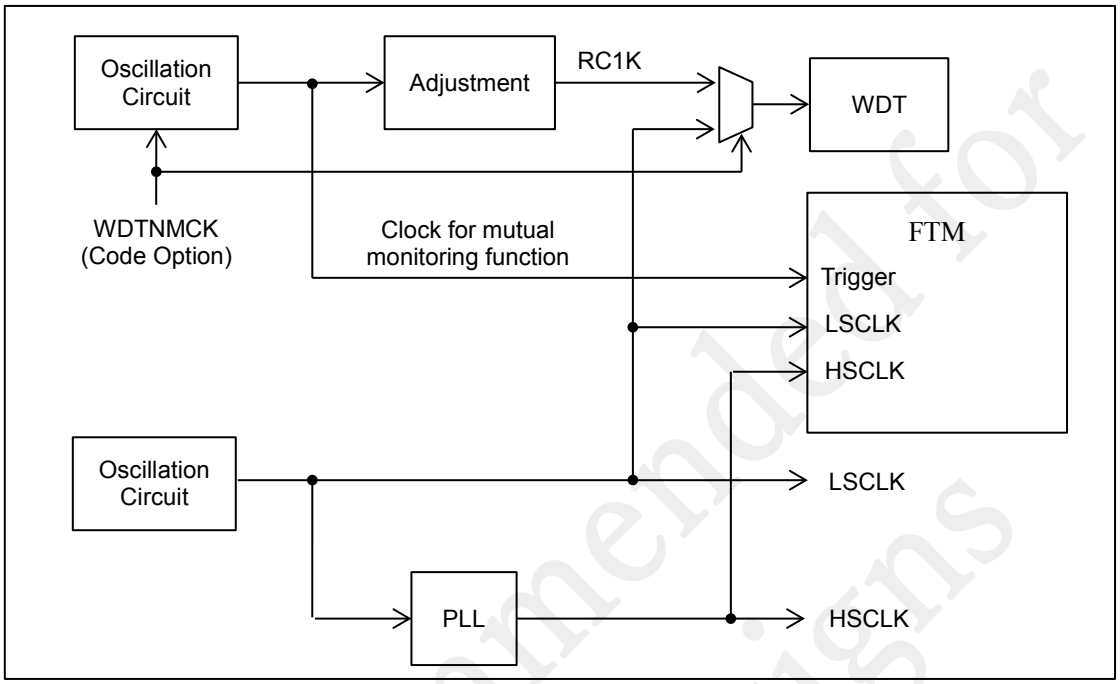
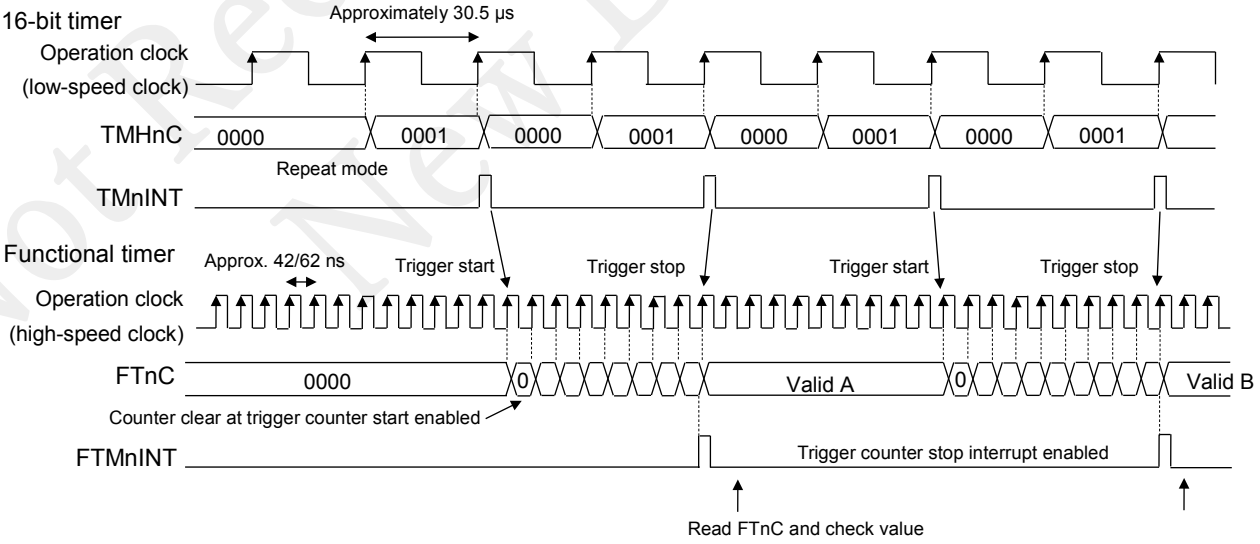


Figure 29-3 Clock Mutual Monitoring Function Block Diagram

Figure 29-4 shows an example of the monitoring operation for the high-speed clock (PLL oscillation circuit) oscillation.



TMHnC: 16-bit timer n counter register
TMHnINT: 16-bit timer n interrupt
FTnC: FTMn counter register
FTMnINT: FTMn interrupt

Figure 29-4 High-Speed Clock (PLL Oscillation Circuit) Oscillation Monitoring Example

Figure 29-5 describes the setting for the monitoring example shown in Figure 29-4.

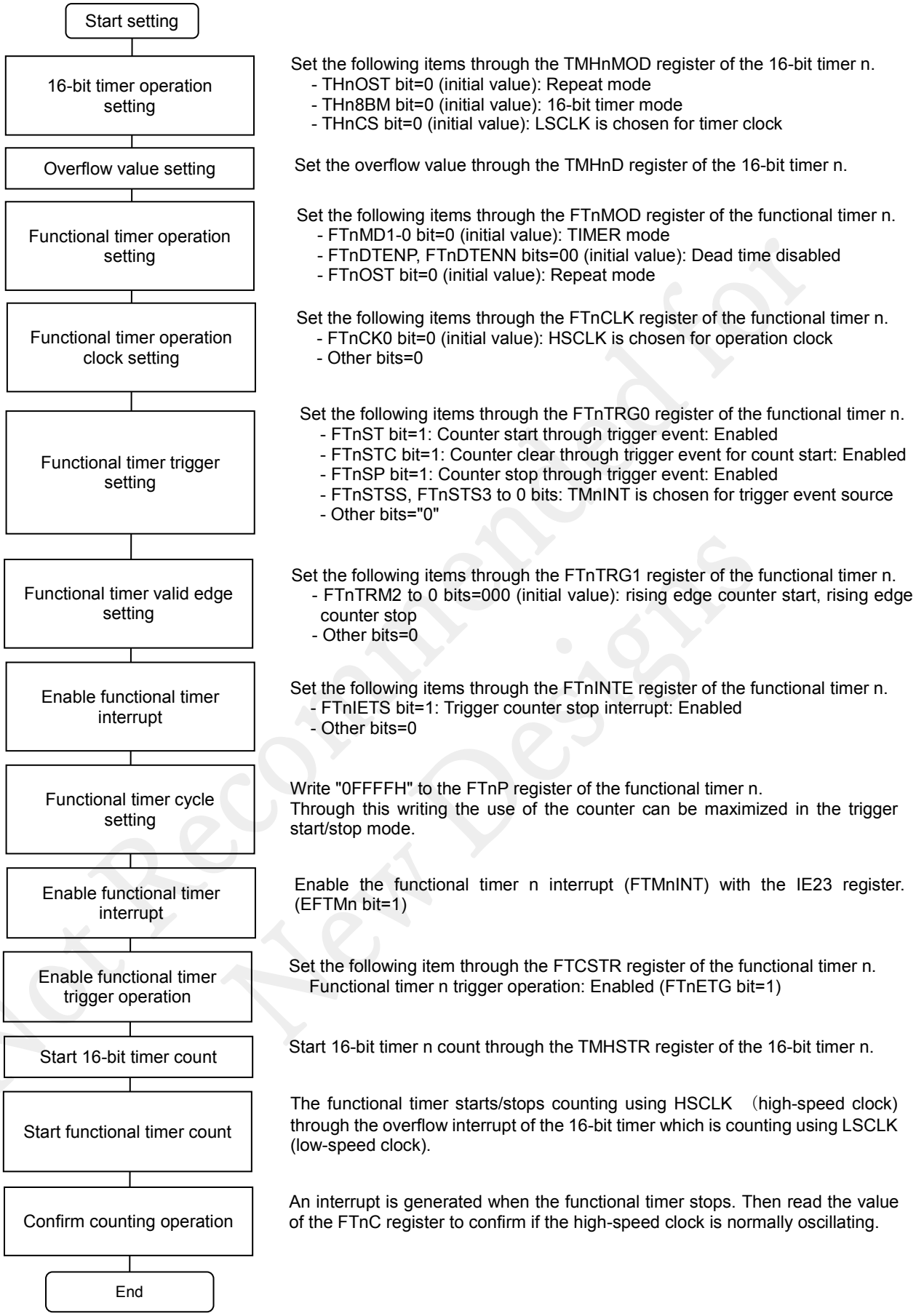


Figure 29-5 Setting of High-Speed Clock (PLL Oscillation Circuit) Oscillation Monitoring Example

[Note]

- **For "Overflow value setting" in Figure 29-5, set the value so that the overflow period of the 16-bit timer n is to be shorter than that of the functional timer n.**
  If the functional timer n overflows, it disables the accurate check. Be careful to prevent overflow of the functional timer n.

29.3.4 CRC Calculation

The CRC (Cyclic Redundancy Check) calculation detects data errors including arbitrary data errors. Two CRC modes are available as described below. Choose one of those depending on the intended use.
See Chapter 19, "CRC Calculator" for details of its operation.

Table 29-1 CRC Calculation Mode

CRC calculation	Description
Automatic CRC calculation mode	Automatically performs calculation of the program code area in units of 32 bits in the HALT/HALT-H mode.
Manual CRC calculation mode	Performs calculation of arbitrary data written from the CPU or DMAC in units of 8 bits.

29.3.5 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.

29.3.6 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.

29.3.7 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 Chapter 23 "Successive Approximation Type A/D Converter" for details.

Appendix A

Appendix A Register List

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
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	0x44
0xF004	Reserved	–	–	–	–	–
0xF005	Reserved	–	–	–	–	–
0xF006	Clock control register	FCON	–	R/W	8	0x00
0xF007	Reserved	–	–	–	–	–
0xF008	High-speed clock wake up time setting register	FHWUPT	–	R/W	8	0x01
0xF009	Reserved	–	–	–	–	–
0xF00A~ 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 L	SBYCONL	SBYCON	W	8	0x00
0xF01B	Standby control register H	SBYCONH	–	R	8	0x00
0xF01C~ 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
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	Reserved	–	–	–	–	–

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
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	GPIO interrupt control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	—	—	—	—	—
0xF047	Reserved	—	-	—	—	—
0xF048	GPIO interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A~ 0xF057	Reserved	—	—	R	8	0x00
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	0x00
0xF05B~ 0xF05F	Reserved	—	—	—	—	—
0xF060	Time base counter register	LTBR	—	R/W	8	0x00
0xF061	Reserved	—	—	—	—	—
0xF062	Time base counter control register	LTBCCON	—	R/W	8	0x01
0xF063H~ 0xF067	Reserved	—	—	—	—	—
0xF068	Time base counter interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x00
0xF069		LTBINTH		R/W	8	0x06
0xF06A~ 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

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
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	Low-speed RC oscillation frequency adjustment register	LRCADJ	–	R/W	8	0x00
0xF081~ 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	Flash data register 1	FLASHD1L	FLASHD1	R/W	8/16	0xFF
0xF095		FLASHD1H		R/W	8	0xFF
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	0x00
0xF0A1~ 0xF0AF	Reserved	–	–	–	–	–
0xF0B0	RAM guard setting register	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	Reserved	–	–	–	–	–
0xF0B9	Reserved	–	–	–	–	–
0xF0BA	SA-ADC test mode register	SADTMOD	–	R/W	8	0x00
0xF0BB	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	Buzzer 0 control register	BZ0CON	–	R/W	8	0x00
0xF0C1	Reserved	–	–	–	–	–
0xF0C2	Buzzer 0 mode register	BZ0MODL	BZ0MOD	R/W	8/16	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF0C3		BZ0MODH		R/W	8	0x00
0xF0C4~ 0xF0CF	Reserved	–	–	–	–	–
0xF0D0	Automatic CRC calculation start address setting register	CRCSADL	CRCSAD	R/W	8/16	0x00
0xF0D1		CRCSADH		R/W	8	0x00
0xF0D2	Automatic CRC calculation end address setting register	CRCEADL	CRCEAD	R/W	8/16	0xFC
0xF0D3		CRCEADH		R/W	8	0xFF
0xF0D4	Automatic CRC calculation start segment setting register	CRCSSEG	–	R/W	8	0x00
0xF0D5	Reserved	–	–	–	–	–
0xF0D6	Automatic CRC calculation end segment setting register	CRCESEG	–	R/W	8	0x0F
0xF0D7	Reserved	–	–	–	–	–
0xF0D8	CRC data register	CRCDATA	–	R/W	8	0x00
0xF0D9	Reserved	–	–	–	–	–
0xF0DA	CRC calculation result register	CRCRESL	CRCRES	R/W	8/16	0xFF
0xF0DB		CRCRESH		R/W	8	0xFF
0xF0DC	Automatic CRC mode register	CRCMOD	–	R/W	8	0x00
0xF0DD	Reserved	–	–	–	–	–
0xF0DE~ 0xF1FF	Reserved	–	–	–	–	–
0xF200	GPIO0 data register	P0DI	P0D	R/W	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	GPIO0 mode register 01	P0MOD0	P0MOD01	R/W	8/16	0x01
0xF203		P0MOD1		R/W	8	0x00
0xF204	GPIO0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x01
0xF205		P0MOD3		R/W	8	0x00
0xF206	GPIO0 mode register 45	P0MOD4	P0MOD45	R/W	8/16	0x00
0xF207		P0MOD5		R/W	8	0x00
0xF208	GPIO0 mode register 67	P0MOD6	P0MOD67	R/W	8/16	0x00
0xF209		P0MOD7		R/W	8	0x00
0xF20A	GPIO0 pulse mode register	P0PMDL	P0PMD	R/W	8/16	0x00
0xF20B		P0PMDH		R/W	8	0x00
0xF20C~ 0xF20F	Reserved	–	–	–	–	–
0xF210	GPIO1 data register	P1DI	P1D	R/W	8/16	0xFF
0xF211		P1DO		R/W	8	0x00
0xF212	GPIO1 mode register 01	P1MOD0	P1MOD01	R/W	8/16	0x00
0xF213		P1MOD1		R/W	8	0x00
0xF214	GPIO1 mode register 23	P1MOD2	P1MOD23	R/W	8/16	0x00
0xF215		P1MOD3		R/W	8	0x00
0xF216	GPIO1 mode register 45	P1MOD4	P1MOD45	R/W	8/16	0x00
0xF217		P1MOD5		R/W	8	0x00
0xF218	GPIO1 mode register 67	P1MOD6	P1MOD67	R/W	8/16	0x00
0xF219		P1MOD7		R/W	8	0x00
0xF21A	GPIO1 pulse mode register	P1PMDL	P1PMD	R/W	8/16	0x00
0xF21B		P1PMDH		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF21C~ 0xF21F	Reserved	–	–	–	–	–
0xF220	GPIO2 data register	P2DI	P2D	R/W	8/16	0xFF
0xF221		P2DO		R/W	8	0x00
0xF222	GPIO2 mode register 01	P2MOD0	P2MOD01	R/W	8/16	0x00
0xF223		P2MOD1		R/W	8	0x00
0xF224	GPIO2 mode register 23	P2MOD2	P2MOD23	R/W	8/16	0x00
0xF225		P2MOD3		R/W	8	0x00
0xF226	GPIO2 mode register 45	P2MOD4	P2MOD45	R/W	8/16	0x00
0xF227		P2MOD5		R/W	8	0x00
0xF228	GPIO2 mode register 67	P2MOD6	P2MOD67	R/W	8/16	0x00
0xF229		P2MOD7		R/W	8	0x00
0xF22A	GPIO2 pulse mode register	P2PMDL	P2PMD	R/W	8/16	0x00
0xF22B		P2PMDH		R/W	8	0x00
0xF22C~ 0xF22F	Reserved	–	–	–	–	–
0xF230	GPIO3 data register	P3DI	P3D	R/W	8/16	0xFF
0xF231		P3DO		R/W	8	0x00
0xF232	GPIO3 mode register 01	P3MOD0	P3MOD01	R/W	8/16	0x00
0xF233		P3MOD1		R/W	8	0x00
0xF234	GPIO3 mode register 23	P3MOD2	P3MOD23	R/W	8/16	0x00
0xF235		P3MOD3		R/W	8	0x00
0xF236	Reserved	-	-	–	–	–
0xF237		-		–	–	–
0xF238		-	-	–	–	–
0xF239		-		–	–	–
0xF23A	GPIO3 pulse mode register	P3PMDL	P3PMD	R/W	8/16	0x00
0xF23B		P3PMDH		R/W	8	0x00
0xF23C~ 0xF2FF	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 1 data register	TMH1DL	TMH1D	R/W	8/16	0xFF
0xF303		TMH1DH		R/W	8	0xFF
0xF304	16－bit timer 2 data register	TMH2DL	TMH2D	R/W	8/16	0xFF
0xF305		TMH2DH		R/W	8	0xFF
0xF306	16－bit timer 3 data register	TMH3DL	TMH3D	R/W	8/16	0xFF
0xF307		TMH3DH		R/W	8	0xFF
0xF308	16－bit timer 4 data register	TMH4DL	TMH4D	R/W	8/16	0xFF
0xF309		TMH4DH		R/W	8	0xFF
0xF30A	16－bit timer 5 data register	TMH5DL	TMH5D	R/W	8/16	0xFF
0xF30B		TMH5DH		R/W	8	0xFF
0xF30C~ 0xF30F	Reserved	–	–	–	–	–
0xF310	16－bit timer 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF311		TMH0CH		R/W	8	0x00
0xF312	16－bit timer 1 counter register	TMH1CL	TMH1C	R/W	8/16	0x00
0xF313		TMH1CH		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF314	16－bit timer 2 counter register	TMH2CL	TMH2C	R/W	8/16	0x00
0xF315		TMH2CH		R/W	8	0x00
0xF316	16－bit timer 3 counter register	TMH3CL	TMH3C	R/W	8/16	0x00
0xF317		TMH3CH		R/W	8	0x00
0xF318	16－bit timer 4 counter register	TMH4CL	TMH4C	R/W	8/16	0x00
0xF319		TMH4CH		R/W	8	0x00
0xF31A	16－bit timer 5 counter register	TMH5CL	TMH5C	R/W	8/16	0x00
0xF31B		TMH5CH		R/W	8	0x00
0xF31C~ 0xF31F	Reserved	－	－	－	－	－
0xF320	16－bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF321		TMH0MODH		R/W	8	0x00
0xF322	16－bit timer 1 mode register	TMH1MODL	TMH1MOD	R/W	8/16	0x00
0xF323		TMH1MODH		R/W	8	0x00
0xF324	16－bit timer 2 mode register	TMH2MODL	TMH2MOD	R/W	8/16	0x00
0xF325		TMH2MODH		R/W	8	0x00
0xF326	16－bit timer 3 mode register	TMH3MODL	TMH3MOD	R/W	8/16	0x00
0xF327		TMH3MODH		R/W	8	0x00
0xF328	16－bit timer 4 mode register	TMH4MODL	TMH4MOD	R/W	8/16	0x00
0xF329		TMH4MODH		R/W	8	0x00
0xF32A	16－bit timer 5 mode register	TMH5MODL	TMH5MOD	R/W	8/16	0x00
0xF32B		TMH5MODH		R/W	8	0x00
0xF32C~ 0xF32F	Reserved	－	－	－	－	－
0xF330	16－bit timer 0 interrupt status register	TMH0ISL	TMH0IS	R/W	8/16	0x00
0xF331		TMH0ISH		R/W	8	0x00
0xF332	16－bit timer 1 interrupt status register	TMH1ISL	TMH1IS	R/W	8/16	0x00
0xF333		TMH1ISH		R/W	8	0x00
0xF334	16－bit timer 2 interrupt status register	TMH2ISL	TMH2IS	R/W	8/16	0x00
0xF335		TMH2ISH		R/W	8	0x00
0xF336	16－bit timer 3 interrupt status register	TMH3ISL	TMH3IS	R/W	8/16	0x00
0xF337		TMH3ISH		R/W	8	0x00
0xF338	16－bit timer 4 interrupt status register	TMH4ISL	TMH4IS	R/W	8/16	0x00
0xF339		TMH4ISH		R/W	8	0x00
0xF33A	16－bit timer 5 interrupt status register	TMH5ISL	TMH5IS	R/W	8/16	0x00
0xF33B		TMH5ISH		R/W	8	0x00
0xF33C~ 0xF33F	Reserved	－	－	－	－	－
0xF340	16－bit timer 0 interrupt clear register	TMH0ICL	TMH0IC	R/W	8/16	0x00
0xF341		TMH0ICH		R/W	8	0x00
0xF342	16－bit timer 1 interrupt clear register	TMH1ICL	TMH1IC	R/W	8/16	0x00
0xF343		TMH1ICH		R/W	8	0x00
0xF344	16－bit timer 2 interrupt clear register	TMH2ICL	TMH2IC	R/W	8/16	0x00
0xF345		TMH2ICH		R/W	8	0x00
0xF346	16－bit timer 3 interrupt clear register	TMH3ICL	TMH3IC	R/W	8/16	0x00
0xF347		TMH3ICH		R/W	8	0x00
0xF348	16－bit timer 4 interrupt clear register	TMH4ICL	TMH4IC	R/W	8/16	0x00
0xF349		TMH4ICH		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF34A	16－bit timer 5 interrupt clear register	TMH5ICL	TMH5IC	R/W	8/16	0x00
0xF34B		TMH5ICH		R/W	8	0x00
0xF34C~ 0xF34F	Reserved	－	－	－	－	－
0xF350	16－bit timer start register	TMHSTRL	TMHSTR	R/W	8/16	0x00
0xF351		TMHSTRH		R/W	8	0x00
0xF352	16－bit timer stop register	TMHSTPL	TMHSTP	R/W	8/16	0x00
0xF353		TMHSTPH		R/W	8	0x00
0xF354	16－bit timer status register	TMHSTATL	TMHSTAT	R/W	8/16	0x00
0xF355		TMHSTATH		R/W	8	0x00
0xF356~ 0xF3FF	Reserved	－	－	－	－	－
0xF400	FTM0 cycle register	FT0PL	FT0P	R/W	8/16	0xFF
0xF401		FT0PH		R/W	8	0xFF
0xF402	FTM1 cycle register	FT1PL	FT1P	R/W	8/16	0xFF
0xF403		FT1PH		R/W	8	0xFF
0xF404	FTM2 cycle register	FT2PL	FT2P	R/W	8/16	0xFF
0xF405		FT2PH		R/W	8	0xFF
0xF406	FTM3 cycle register	FT3PL	FT3P	R/W	8/16	0xFF
0xF407		FT3PH		R/W	8	0xFF
0xF408~ 0xF40F	Reserved	－	－	－	－	－
0xF410	FTM0 event A register	FT0EAL	FT0EA	R/W	8/16	0x00
0xF411		FT0EAH		R/W	8	0x00
0xF412	FTM1 event A register	FT1EAL	FT1EA	R/W	8/16	0x00
0xF413		FT1EAH		R/W	8	0x00
0xF414	FTM2 event A register	FT2EAL	FT2EA	R/W	8/16	0x00
0xF415		FT2EAH		R/W	8	0x00
0xF416	FTM3 event A register	FT3EAL	FT3EA	R/W	8/16	0x00
0xF417		FT3EAH		R/W	8	0x00
0xF418~ 0xF41F	Reserved	－	－	－	－	－
0xF420	FTM0 event B register	FT0EBL	FT0EB	R/W	8/16	0x00
0xF421		FT0EBH		R/W	8	0x00
0xF422	FTM1 event B register	FT1EBL	FT1EB	R/W	8/16	0x00
0xF423		FT1EBH		R/W	8	0x00
0xF424	FTM2 event B register	FT2EBL	FT2EB	R/W	8/16	0x00
0xF425		FT2EBH		R/W	8	0x00
0xF426	FTM3 event B register	FT3EBL	FT3EB	R/W	8/16	0x00
0xF427		FT3EBH		R/W	8	0x00
0xF428~ 0xF42F	Reserved	－	－	－	－	－
0xF430	FTM0 dead time register	FT0DTL	FT0DT	R/W	8/16	0x00
0xF431		FT0DTH		R/W	8	0x00
0xF432	FTM1 dead time register	FT1DTL	FT1DT	R/W	8/16	0x00
0xF433		FT1DTH		R/W	8	0x00
0xF434	FTM2 dead time register	FT2DTL	FT2DT	R/W	8/16	0x00
0xF435		FT2DTH		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF436	FTM3 dead time register	FT3DTL	FT3DT	R/W	8/16	0x00
0xF437		FT3DTH		R/W	8	0x00
0xF438~ 0xF43F	Reserved	–	–	–	–	–
0xF440	FTM0 counter register	FT0CL	FT0C	R/W	8/16	0x00
0xF441		FT0CH		R/W	8	0x00
0xF442	FTM1 counter register	FT1CL	FT1C	R/W	8/16	0x00
0xF443		FT1CH		R/W	8	0x00
0xF444	FTM2 counter register	FT2CL	FT2C	R/W	8/16	0x00
0xF445		FT2CH		R/W	8	0x00
0xF446	FTM3 counter register	FT3CL	FT3C	R/W	8/16	0x00
0xF447		FT3CH		R/W	8	0x00
0xF448~ 0xF44F	Reserved	–	–	–	–	–
0xF450	FTM0 status register	FT0STAT	–	R/W	8	0x00
0xF451	Reserved	–	–	R	8	0x00
0xF452	FTM1 status register	FT1STAT	–	R/W	8	0x00
0xF453	Reserved	–	–	R	8	0x00
0xF454	FTM2 status register	FT2STAT	–	R/W	8	0x00
0xF455	Reserved	–	–	R	8	0x00
0xF456	FTM3 status register	FT3STAT	–	R/W	8	0x00
0xF457	Reserved	–	–	R	8	0x00
0xF458~ 0xF45F	Reserved	–	–	–	–	–
0xF460	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF461		FT0MODH		R/W	8	0x00
0xF462	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF463		FT1MODH		R/W	8	0x00
0xF464	FTM2 mode register	FT2MODL	FT2MOD	R/W	8/16	0x00
0xF465		FT2MODH		R/W	8	0x00
0xF466	FTM3 mode register	FT3MODL	FT3MOD	R/W	8/16	0x00
0xF467		FT3MODH		R/W	8	0x00
0xF468~ 0xF46F	Reserved	–	–	–	–	–
0xF470	FTM0 clock register	FT0CLKL	FT0CLK	R/W	8/16	0x00
0xF471		FT0CLKH		R/W	8	0x00
0xF472	FTM1 clock register	FT1CLKL	FT1CLK	R/W	8/16	0x00
0xF473		FT1CLKH		R/W	8	0x00
0xF474	FTM2 clock register	FT2CLKL	FT2CLK	R/W	8/16	0x00
0xF475		FT2CLKH		R/W	8	0x00
0xF476	FTM3 clock register	FT3CLKL	FT3CLK	R/W	8/16	0x00
0xF477		FT3CLKH		R/W	8	0x00
0xF478~ 0xF47F	Reserved	–	–	–	–	–
0xF480	FTM0 trigger register 0	FT0TRG0L	FT0TRG0	R/W	8/16	0x00
0xF481		FT0TRG0H		R/W	8	0x00
0xF482	FTM1 trigger register 0	FT1TRG0L	FT1TRG0	R/W	8/16	0x00
0xF483		FT1TRG0H		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF484	FTM2 trigger register 0	FT2TRG0L	FT2TRG0	R/W	8/16	0x00
0xF485		FT2TRG0H		R/W	8	0x00
0xF486	FTM3 trigger register 0	FT3TRG0L	FT3TRG0	R/W	8/16	0x00
0xF487		FT3TRG0H		R/W	8	0x00
0xF488~ 0xF48F	Reserved	–	–	–	–	–
0xF490	FTM0 trigger register 1	FT0TRG1L	FT0TRG1	R/W	8/16	0x00
0xF491		FT0TRG1H		R/W	8	0x00
0xF492	FTM1 trigger register 1	FT1TRG1L	FT1TRG1	R/W	8/16	0x00
0xF493		FT1TRG1H		R/W	8	0x00
0xF494	FTM2 trigger register 1	FT2TRG1L	FT2TRG1	R/W	8/16	0x00
0xF495		FT2TRG1H		R/W	8	0x00
0xF496	FTM3 trigger register 1	FT3TRG1L	FT3TRG1	R/W	8/16	0x00
0xF497		FT3TRG1H		R/W	8	0x00
0xF498~ 0xF49F	Reserved	–	–	–	–	–
0xF4A0	FTM0 interrupt enable register	FT0INTEL	FT0INTE	R/W	8/16	0x00
0xF4A1		FT0INTEH		R/W	8	0x00
0xF4A2	FTM1 interrupt enable register	FT1INTEL	FT1INTE	R/W	8/16	0x00
0xF4A3		FT1INTEH		R/W	8	0x00
0xF4A4	FTM2 interrupt enable register	FT2INTEL	FT2INTE	R/W	8/16	0x00
0xF4A5		FT2INTEH		R/W	8	0x00
0xF4A6	FTM3 interrupt enable register	FT3INTEL	FT3INTE	R/W	8/16	0x00
0xF4A7		FT3INTEH		R/W	8	0x00
0xF4A8~ 0xF4AF	Reserved	–	–	–	–	–
0xF4B0	FTM0 interrupt status register	FT0INTSL	FT0INTS	R	8/16	0x00
0xF4B1		FT0INTSH		R	8	0x00
0xF4B2	FTM1 interrupt status register	FT1INTSL	FT1INTS	R	8/16	0x00
0xF4B3		FT1INTSH		R	8	0x00
0xF4B4	FTM2 interrupt status register	FT2INTSL	FT2INTS	R	8/16	0x00
0xF4B5		FT2INTSH		R	8	0x00
0xF4B6	FTM3 interrupt status register	FT3INTSL	FT3INTS	R	8/16	0x00
0xF4B7		FT3INTSH		R	8	0x00
0xF4B8~ 0xF4BF	Reserved	–	–	–	–	–
0xF4C0	FTM0 interrupt clear register	FT0INTCL	FT0INTC	W	8/16	0x00
0xF4C1		FT0INTCH		W	8	0x00
0xF4C2	FTM1 interrupt clear register	FT1INTCL	FT1INTC	W	8/16	0x00
0xF4C3		FT1INTCH		W	8	0x00
0xF4C4	FTM2 interrupt clear register	FT2INTCL	FT2INTC	W	8/16	0x00
0xF4C5		FT2INTCH		W	8	0x00
0xF4C6	FTM3 interrupt clear register	FT3INTCL	FT3INTC	W	8/16	0x00
0xF4C7		FT3INTCH		W	8	0x00
0xF4C8~ 0xF4EF	Reserved	–	–	–	–	–
0xF4F0	FTM common update register	FTCUD	–	W	8	0x00
0xF4F1	Reserved	–	–	–	–	–

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF4F2	FTM common control register L	FTCCONL	FTCCON	W/R	8/16	0x00
0xF4F3	FTM common control register H	FTCCONH		W/R	8	0x00
0xF4F4	FTM common start register L	FTCSTRL	FTCSTR	W	8/16	0x00
0xF4F5	FTM common start register H	FTCSTRH		W	8	0x00
0xF4F6	FTM common stop register L	FCSTPL	FCSTR	W	8/16	0x00
0xF4F7	FTM common stop register H	FCSTPH		W	8	0x00
0xF4F8	FTM common status register L	FTCSTATL	FTCSTA	R	8	0x00
0xF4F9	FTM common status register H	FTCSTATH		R	8	0x00
0xF4FA~ 0xF5FF	Reserved	–	–	–	–	–
0xF600	Serial communication unit 0 transmit/receive buffer	SD0BUFL	SD0BUF	R/W	8/16	0x00
0xF601		SD0BUFH		R/W	8	0x00
0xF602	Serial communication unit 0 mode register	SU0MOD	–	R/W	8	0x00
0xF603		–		R	8	0x00
0xF604	Serial communication unit 0 transmission interval setting register	SU0DLYL	SU0DLY	R/W	8	0x00
0xF605		–		R	8	0x00
0xF606	Serial communication unit 0 control register	SU0CONL	SIO0MOD	R/W	8	0x00
0xF607		SU0CONH		R	8	0x00
0xF608	Synchronous serial port 0 mode register	SIO0MODL	SIO0MOD	R/W	8/16	0x00
0xF609		SIO0MODH		R/W	8	0x00
0xF60A	Synchronous serial port 0 status register	SIO0STAT	–	R/W	8	0x00
0xF60B	Reserved	–	–	–	–	–
0xF60C	UART00 mode register	UA00MODL	UA00MOD	R/W	8/16	0x00
0xF60D		UA00MODH		R/W	8	0x00
0xF60E	UART00 baud rate register	UA00BRTL	UA00BRT	R/W	8/16	0xFF
0xF60F		UA00BRTH		R/W	8	0xFF
0xF610	UART00 baud rate adjustment register	UA00BRC	–	R/W	8	0x00
0xF611	Reserved	–	–	–	–	–
0xF612	UART00 status register	UA00STAT	–	R/W	8	0x00
0xF613	Reserved	–	–	–	–	–
0xF614	UART01 mode register	UA01MODL	UA01MOD	R/W	8/16	0x00
0xF615		UA01MODH		R/W	8	0x00
0xF616	UART01 baud rate register	UA01BRTL	UA01BRT	R/W	8/16	0xFF
0xF617		UA01BRTH		R/W	8	0xFF
0xF618	UART01 baud rate adjustment register	UA01BRC	–	R/W	8	0x00
0xF619	Reserved	–	–	–	–	–
0xF61A	UART01 status register	UA01STAT	–	R/W	8	0x00
0xF61B~ 0xF61F	Reserved	–	–	–	–	–
0xF620	Serial communication unit 1 transmit/receive buffer	SD1BUFL	SD1BUF	R/W	8/16	0x00
0xF621		SD1BUFH		R/W	8	0x00
0xF622	Serial communication unit 1 mode register	SU1MOD	–	R/W	8	0x00
0xF623	Reserved	–	–	–	–	–
0xF624	Serial communication unit 1 transmission interval setting register	SU1DLYL	SU1DLY	R/W	8/16	0x00
0xF625		–		R	8	0x00
0xF626	Serial communication unit 1 control register	SU1CONL	SU1CON	R/W	8/16	0x00
0xF627		SU1CONH		R/W	8	0x00
0xF628	Synchronous serial port 1 mode register	SIO1MODL	SIO1MOD	R/W	8/16	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF629		SIO1MODH		R/W	8	0x00
0xF62A	Synchronous serial port 1 status register	SIO1STAT	—	R/W	8	0x00
0xF62B	Reserved	—	—	—	—	—
0xF62C	UART10 mode register	UA10MODL	UA10MOD	R/W	8/16	0x00
0xF62D		UA10MODH		R/W	8	0x00
0xF62E	UART10 baud rate register	UA10BRTL	UA10BRT	R/W	8/16	0xFF
0xF62F		UA10BRTH		R/W	8	0xFF
0xF630	UART10 baud rate adjustment register	UA10BRC	—	R/W	8	0x00
0xF631	Reserved	—	—	—	—	—
0xF632	UART10 status register	UA10STAT	—	R/W	8	0x00
0xF633	Reserved	—	—	—	—	—
0xF634	UART11 mode register	UA11MODL	UA11MOD	R/W	8/16	0x00
0xF635		UA11MODH		R/W	8	0x00
0xF636	UART11 baud rate register	UA11BRTL	UA11BRT	R/W	8/16	0xFF
0xF637		UA11BRTH		R/W	8	0xFF
0xF638	UART11 baud rate adjustment register	UA11BRC	—	R/W	8	0x00
0xF639	Reserved	—	—	—	—	—
0xF63A	UART11 status register	UA11STAT	—	R/W	8	0x00
0xF63B	Reserved	—	—	—	—	—
0xF63C~ 0xF6BF	Reserved	—	—	—	—	—
0xF6C0	I^2C bus unit 0 master/slave select	I2U0MSS	—	R/W	8	0x00
0xF6C1	Reserved	—	—	—	—	—
0xF6C2	I2C bus unit 0 receive register (master)	I2UM0RD	—	R	8	0x00
0xF6C3	Reserved	—	—	—	—	—
0xF6C4	I2C bus 0 slave address register (master)	I2UM0SA	—	R/W	8	0x00
0xF6C5	Reserved	—	—	—	—	—
0xF6C6	I2C bus 0 transmit data register (master)	I2UM0TD	—	R/W	8	0x00
0xF6C7	Reserved	—	—	R	8	0x00
0xF6C8	I2C bus 0 control register (master)	I2UM0CON	—	R/W	8	0x00
0xF6C9	Reserved	—	—	—	—	—
0xF6CA	I2C bus 0 mode register (master)	I2UM0MDL	I2UM0MOD	R/W	8/16	0x00
0xF6CB		I2UM0MDH		R/W	8	0x02
0xF6CC	I2C bus 0 status register (master)	I2UM0STA	—	R/W	8	0x00
0xF6CD	Reserved	—	—	—	—	—
0xF6CE	I2C bus 0 receive register (slave)	I2US0RD	—	R	8	0x00
0xF6CF	Reserved	—	—	—	—	—
0xF6D0	I2C bus 0 slave address register (slave)	I2US0SA	—	R/W	8	0x00
0xF6D1	Reserved	—	—	—	—	—
0xF6D2	I2C bus 0 transmit data register (slave)	I2US0TD	—	R/W	8	0x00
0xF6D3	Reserved	—	—	—	—	—
0xF6D4	I2C bus 0 control register (slave)	I2US0CON	—	R/W	8	0x00
0xF6D5	Reserved	—	—	—	—	—
0xF6D6	I2C bus 0 mode register (slave)	I2US0MD	—	R/W	8	0x00
0xF6D7	Reserved	—	—	—	—	—
0xF6D8	I2C bus 0 status register (slave)	I2US0STA	—	R/W	8	0x00
0xF6D9	Reserved	—	—	—	—	—

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF6DA~ 0xF6E1	Reserved	–	–	–	–	–
0xF6E2	I^2C master 0 receive register	I2M0RD	–	R	8	0x00
0xF6E3	Reserved	–	–	–	–	–
0xF6E4	I^2C master 0 slave address register	I2M0SA	–	R/W	8	0x00
0xF6E5	Reserved	–	–	–	–	–
0xF6E6	I^2C master 0 transmit data register	I2M0TD	–	R/W	8	0x00
0xF6E7	Reserved	–	–	–	–	–
0xF6E8	I^2C master 0 control register	I2M0CON	–	R/W	8	0x00
0xF6E9	Reserved	–	–	–	–	–
0xF6EA	I^2C master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF6EB		I2M0MODH		R/W	8	0x02
0xF6EC	I^2C master 0 status register	I2M0STAT	–	R/W	8	0x00
0xF6ED~ 0xF6FF	Reserved	–	–	–	–	–
0xF700	DMA channel 0 transfer mode register	DC0MODL	DC0MOD	R/W	8/16	0x00
0xF701		DC0MODH		R/W	8	0x00
0xF702	DMA channel 0 transfer count register	DC0TNL	DC0TN	R/W	8/16	0x00
0xF703		DC0TNH		R/W	8	0x00
0xF704	DMA channel 0 transfer source address register	DC0SAL	DC0SA	R/W	8/16	0x00
0xF705		DC0SAH		R/W	8	0x00
0xF706	DMA channel 0 transfer destination address register	DC0DAL	DC0DA	R/W	8/16	0x00
0xF707		DC0DAH		R/W	8	0x00
0xF708	DMA channel 1 transfer mode register	DC1MODL	DC1MOD	R/W	8/16	0x00
0xF709		DC1MODH		R/W	8	0x00
0xF70A	DMA channel 1 transfer count register	DC1TNL	DC1TN	R/W	8/16	0x00
0xF70B		DC1TNH		R/W	8	0x00
0xF70C	DMA channel 1 transfer source start address register	DC1SAL	DC1SA	R/W	8/16	0x00
0xF70D		DC1SAH		R/W	8	0x00
0xF70E	DMA channel 1 transfer destination start address register	DC1DAL	DC1DA	R/W	8/16	0x00
0xF70F		DC1DAH		R/W	8	0x00
0xF710~ 0xF71F	Reserved	–	–	–	–	–
0xF720	DMA transfer enable register	DCEN	–	R/W	8	0x00
0xF721	Reserved	–	–	–	–	–
0xF722	DMA status register	DSTATL	DSTAT	R/	8	0x00
0xF723		DSTATH		R	8	0x00
0xF724	DMA interrupt status clear register	DICLR	–	W	8	0x00
0xF725	Reserved	–	–	–	–	–
0xF726~ 0xF7FF	Reserved	–	–	–	–	–
0xF800	SA－ADC result register 0	SADR0L	SADR0	R	8/16	0x00
0xF801		SADR0H		R	8	0x00
0xF802	SA－ADC result register 1	SADR1L	SADR1	R	8/16	0x00
0xF803		SADR1H		R	8	0x00
0xF804	SA－ADC result register 2	SADR2L	SADR2	R	8/16	0x00
0xF805		SADR2H		R	8	0x00
0xF806	SA－ADC result register 3	SADR3L	SADR3	R	8/16	0x00

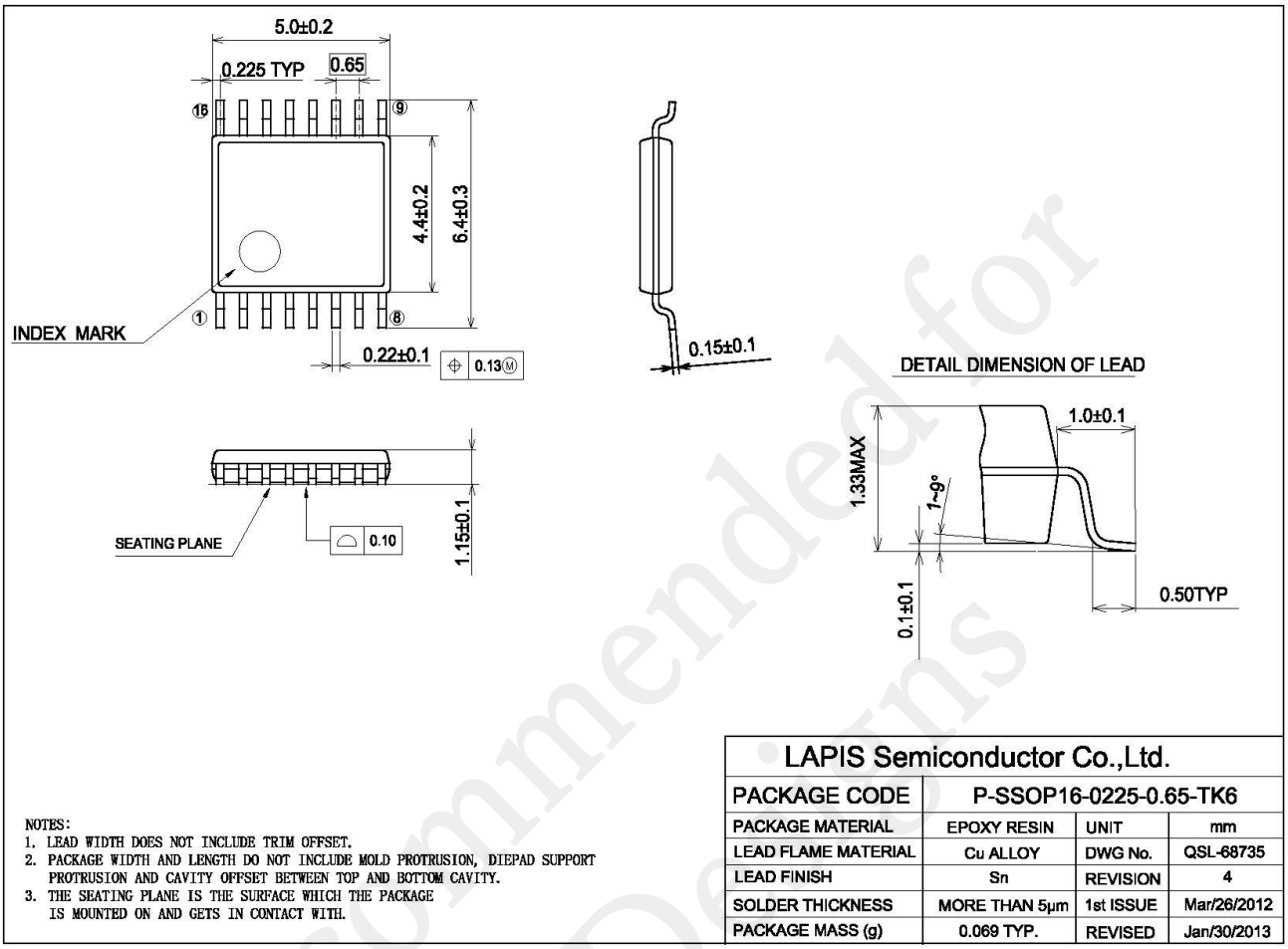
Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF807		SADR3H		R	8	0x00
0xF808	SA−ADC result register 4	SADR4L	SADR4	R	8/16	0x00
0xF809		SADR4H		R	8	0x00
0xF80A	SA−ADC result register 5	SADR5L	SADR5	R	8/16	0x00
0xF80B		SADR5H		R	8	0x00
0xF80C	SA−ADC result register 6	SADR6L	SADR6	R	8/16	0x00
0xF80D		SADR6H		R	8	0x00
0xF80E	SA−ADC result register 7	SADR7L	SADR7	R	8/16	0x00
0xF80F		SADR7H		R	8	0x00
0xF810~ 0xF81F	Reserved	−	−	−	−	−
0xF820	SA−ADC result register 10	SADR10L	SADR10	R	8/16	0x00
0xF821		SADR10H		R	8	0x00
0xF822	SA−ADC result register	SADRL	SADR	R	8/16	0x00
0xF823		SADRH		R	8	0x00
0xF824	SA−ADC upper/lower limit status register 0	SADULS0L	SADULS0	R/W	8/16	0x00
0xF825		SADULS0H		R/W	8	0x00
0xF826	SA−ADC upper/lower limit status register 1	SADULS1L	SADULS1	R/W	8/16	0x00
0xF827		SADULS1H		R/W	8	0x00
0xF828	SA−ADC mode register	SADMODL	SADMOD	R/W	8/16	0x00
0xF829		SADMODH		R/W	8	0x00
0xF82A	SA−ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF82B		SADCONH		R/W	8	0x00
0xF82C	SA−ADC enable register 0	SADEN0L	SADE0N	R/W	8/16	0x00
0xF82D		SADEN0H		R/W	8	0x00
0xF82E	SA−ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF82F		SADEN1H		R/W	8	0x00
0xF830	Reserved	−	−	−	−	−
0xF831	Reserved	−	−	−	−	−
0xF832	SA−ADC scan conversion interval setting register	SADSTML	SADSTM	R/W	8/16	0x00
0xF833		SADSTMH		R/W	8	0x00
0xF834	SA−ADC conversion result mode setting register	SADLMODL	SADLMOD	R/W	8/16	0x00
0xF835		SADLMODH		R/W	8	0x00
0xF836	SA−ADC conversion result upper limit setting register	SADUPLL	SADUPL	R/W	8/16	0xC0
0xF837		SADUPLH		R/W	8	0xFF
0xF838	SA−ADC conversion result lower limit setting register	SADLOLL	SADLOL	R/W	8/16	0x00
0xF839		SADLOLH		R/W	8	0x00
0xF83A	SA−ADC TEMP/VREF control register	VREFCON	−	R/W	8	0x00
0xF83B	Reserved	−	−	−	−	−
0xF83C	SA−ADC interrupt mode register	SADIMOD	−	R/W	8	0x00
0xF83D	Reserved	−	−	−	−	−
0xF83E	SA−ADC trigger register	SADTRG	−	R/W	8	0x00
0xF83F	Reserved	−	−	−	−	−
0xF840	Comparator 0 control register	CMP0CON	−	R/W	8	0x00
0xF841	Reserved	−	−	−	−	−
0xF842	Comparator 0 mode register	CMP0MODL	CMP0MOD	R/W	8/16	0x00
0xF843		CMP0MODH		R/W	8	0x00

Address [H]	Name	Symbol (Byte)	Symbol (Word)	R/W	Size	Initial value[H]
0xF844~ 0xF84F	Reserved	–	–	–	–	–
0xF850	Voltage level supervisor 0 control register	VLS0CON	–	R/W	8	0x00
0xF851	Reserved	–	–	–	–	–
0xF852	Voltage level supervisor 0 mode register	VLS0MOD	–	R/W	8	0x00
0xF853	Reserved	–	–	–	–	–
0xF854	Voltage level supervisor 0 level register	VLS0LV	–	R/W	8	0x00
0xF855	Reserved	–	–	–	–	–
0xF856	Voltage level supervisor 0 sampling register	VLS0SMP	–	R/W	8	0x00
0xF857	Reserved	–	–	–	–	–
0xF858~ 0xF85F	Reserved	–	–	–	–	–
0xF860	D/A converter control register	DACCON	–	R/W	8	0x00
0xF861	Reserved	–	–	–	–	–
0xF862	D/A converter code register	DACCODE	–	R/W	8	0x00
0xF863~ 0xF8FF	Reserved	–	–	–	–	–
0xF900~ 0xFFFF	Reserved	–	–	–	–	–

Appendixes

Appendix B. Package Dimensions

ML62Q1223A/1224A/1225A16pin SSOP



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.

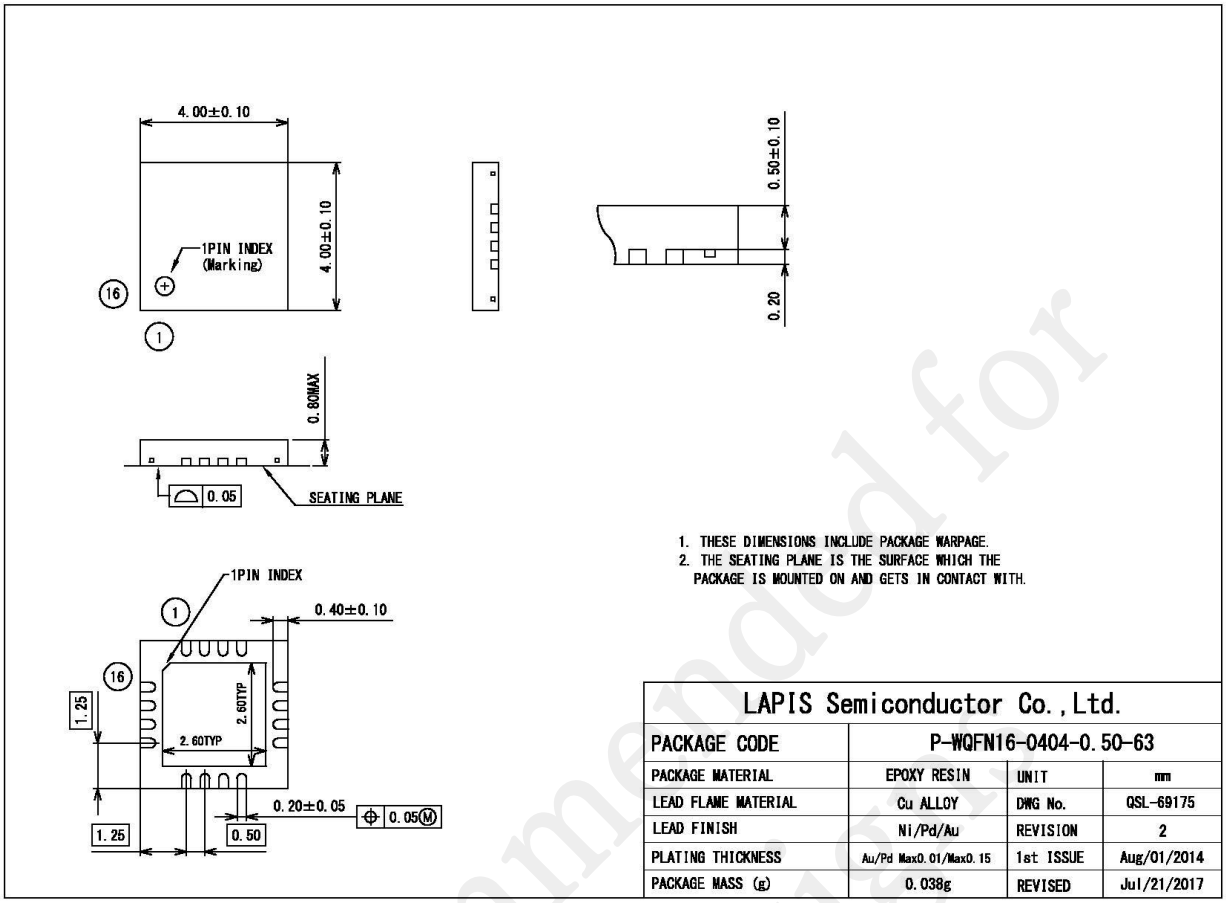
LAPIS Semiconductor Co.,Ltd.			
PACKAGE CODE	P-SSOP16-0225-0.65-TK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-68735
LEAD FINISH	Sn	REVISION	4
SOLDER THICKNESS	MORE THAN 5μm	1st ISSUE	Mar/26/2012
PACKAGE MASS (g)	0.069 TYP.	REVISED	Jan/30/2013

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).

ML62Q1223A/1224A/1225A16pin WQFN



LAPIS Semiconductor Co.,Ltd.			
PACKAGE CODE	P-WQFN16-0404-0.50-63		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69175
LEAD FINISH	Ni/Pd/Au	REVISION	2
PLATING THICKNESS	Au/Pd Max0.01/Max0.15	1st ISSUE	Aug/01/2014
PACKAGE MASS (g)	0.038g	REVISED	Jul/21/2017

Notes for Mounting the Surface Mount Type Package

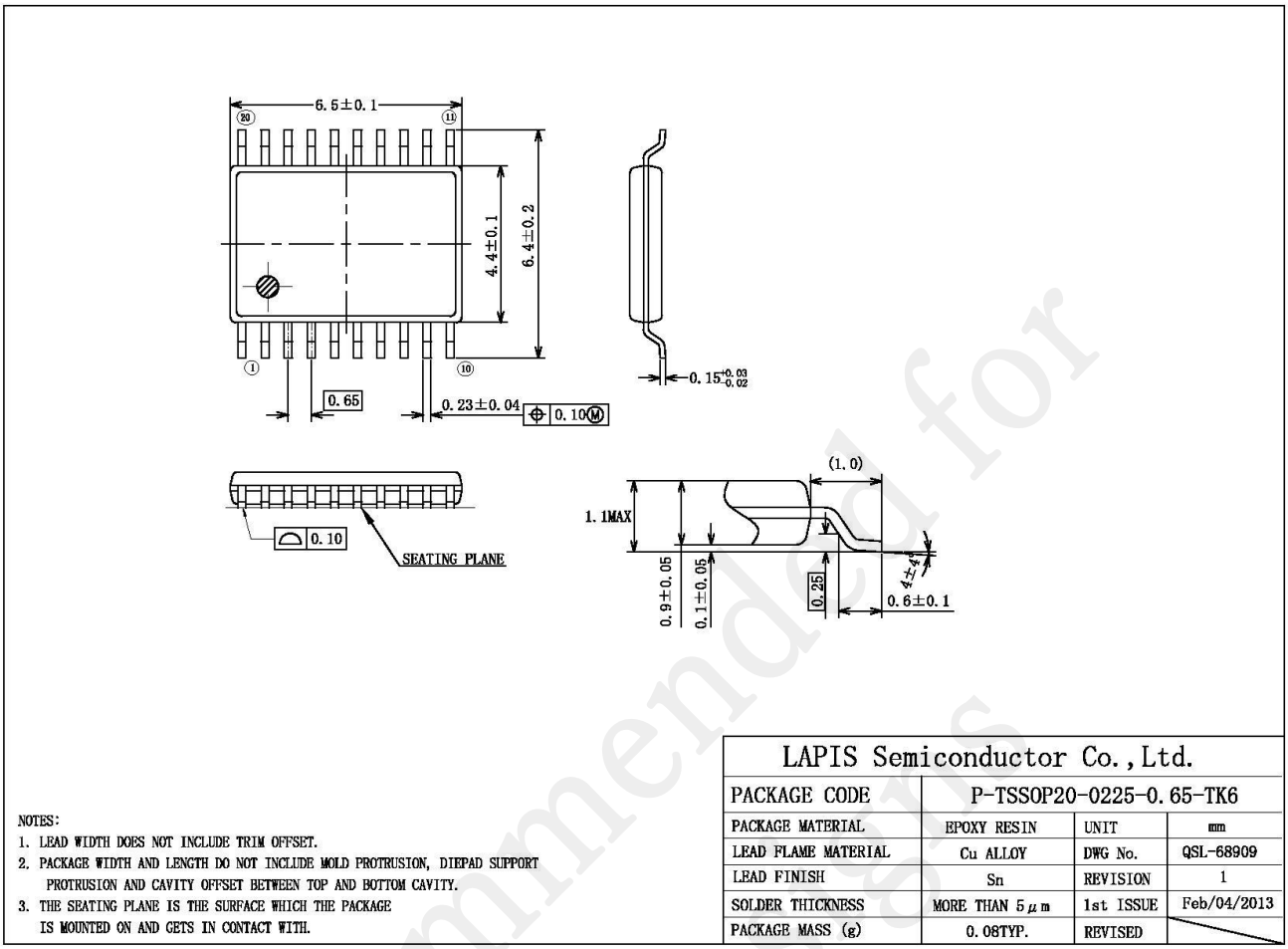
The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage.

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 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.

ML62Q1233A/1234A/1235A20pin TSSOP



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.

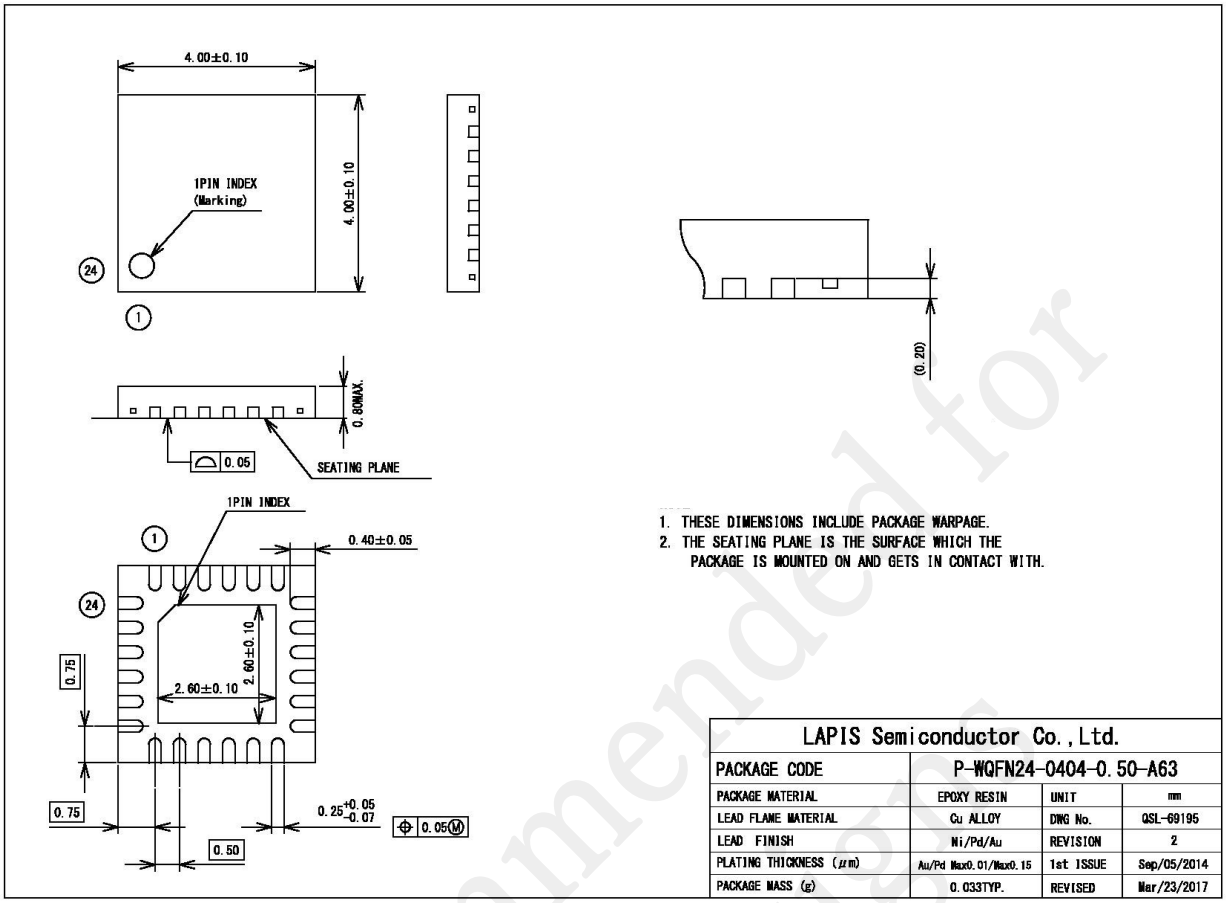
LAPIS Semiconductor Co.,Ltd.			
PACKAGE CODE	P-TSSOP20-0225-0.65-TK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-68909
LEAD FINISH	Sn	REVISION	1
SOLDER THICKNESS	MORE THAN 5μm	1st ISSUE	Feb/04/2013
PACKAGE MASS (g)	0.08TYP.	REVISED	

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).

ML62Q1245A/1246A/1247A24pin WQFN



LAPIS Semiconductor Co.,Ltd.			
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	2
PLATING THICKNESS (μm)	Au/Pd Max0.01/Max0.15	1st ISSUE	Sep/05/2014
PACKAGE MASS (g)	0.033TYP.	REVISED	Mar/23/2017

Notes for Mounting the Surface Mount Type Package

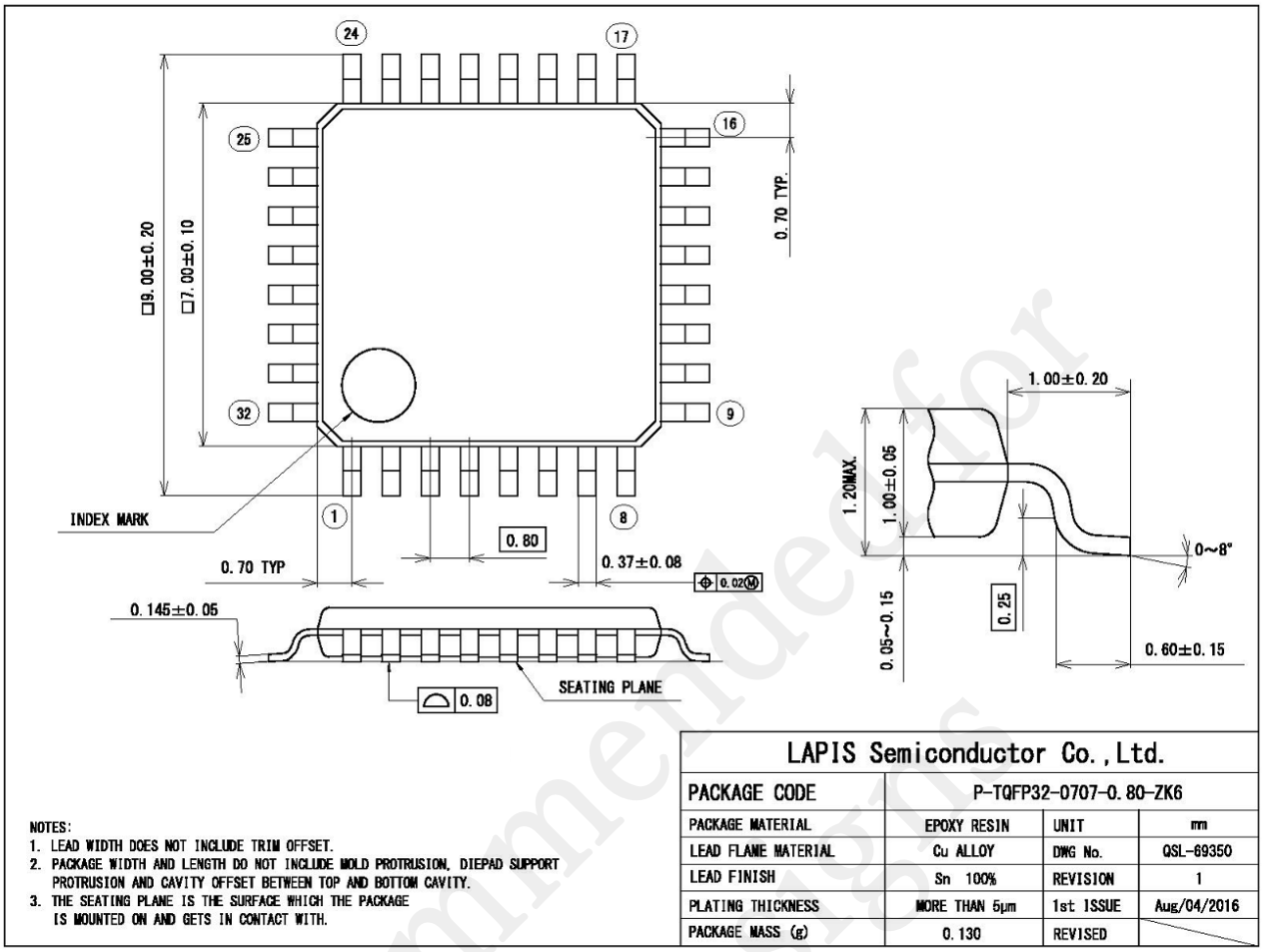
The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage.

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 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.

ML62Q1265A/1266A/1267A32pin TQFP



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.

LAPIS Semiconductor Co.,Ltd.			
PACKAGE CODE	P-TQFP32-0707-0.80-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69350
LEAD FINISH	Sn 100%	REVISION	1
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Aug/04/2016
PACKAGE MASS (g)	0.130	REVISED	

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).

Appendix C. Instruction Execution Cycle

ML62Q1000 series has two CPU operating modes defined as the no wait mode and wait mode, in which there are some cases the insturction execution cycles gets different.

- No wait mode
  There is no instruction execution cycle increases.

- Wait mode
  Some instruction execution cycles increase in the case wait cycles are required for reading the program memory during the execution.

Tables on following pages show the all instructions of nX-U16/100 core and the execution cycles in the two CPU modes. "-" indicates that threre 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)-1	(2)-2	(3)-1	(3)-2	(4)	(5)
Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
ADD	ER*n*	ER*m*	1	1	-	-	-	-
B	*Cadr*	-	2	6	-	-	-	1
-	ER*n*	-	2	6	-	-	-	1
L	ER*n*	[EA]	1	1	1	5	1	-
-	ER*n*	[EA+]	1	1	1	5	1	-

[How to read the table]

(1) These are the instructions of nX-U16/100(A35 core)
(2) The instruction execution cycle in the case of no wait mode and wait mode.
(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+] addressig.
    One cycle is added in spite of the CPU operating mode.
    For more details, see the section 3.3 "Instruction Execution Times" in the nX-U16/100 core instruction manual.

Arithmetic Instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
ADD	ER*n*	ER*m*	1	1	-	-	-	-
		#imm7	1	1	-	-	-	-
ADD	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
ADDC	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
AND	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
CMP	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
CMPC	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
MOV	ERn	ER*m*	1	1	-	-	-	-
		#imm7	1	1	-	-	-	-
MOV	Rn	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
OR	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
XOR	R*n*	R*m*	1	1	-	-	-	-
		#imm8	1	1	-	-	-	-
CMP	ER*n*	ER*m*	1	1	-	-	-	-
SUB	R*n*	R*m*	1	1	-	-	-	-
SUBC	R*n*	R*m*	1	1	-	-	-	-

Shift instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
SLL	R*n*	R*m*	1	1	-	-	-	1
		#width	1	1	-	-	-	1
SLLC	R*n*	R*m*	1	1	-	-	-	1
		#width	1	1	-	-	-	1
SRA	R*n*	R*m*	1	1	-	-	-	1
		#width	1	1	-	-	-	1
SRL	R*n*	R*m*	1	1	-	-	-	1
		#width	1	1	-	-	-	1
SRLC	R*n*	R*m*	1	1	-	-	-	1
		#width	1	1	-	-	-	1

Load/Store instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
L	ER*n*	[EA]	1	1	1	5	1	-
		[EA+]	1	1	1	5	1	-
		[ER*m*]	1	1 or 2	1	5	1	1
		Disp16[ER*m*]	2	2	1	5	1	1
		Disp6[BP]	2	2	1	5	1	1
		Disp6[FP]	2	2	1	5	1	1
		Dadr	2	2	1	5	1	1
	R*n*	[EA]	1	1	1	5	1	-
		[EA+]	1	1	1	5	1	-
		[ER*m*]	1	1 or 2	1	5	1	1
		Disp16[ER*m*]	2	2	1	5	1	1
		Disp6[BP]	2	2	1	5	1	1
		Disp6[FP]	2	2	1	5	1	1
		Dadr	2	2	1	5	1	1
	XR*n*	[EA]	2	2	2	10	1	-
		[EA+]	2	2	2	10	1	-
	QR*n*	[EA]	4	4	4	15	1	-
		[EA+]	4	4	4	15	1	-
ST	ER*n*	[EA]	1	1	-	-	-	-
		[EA+]	1	1	-	-	-	-
		[ER*m*]	1	1 or 2	-	-	-	1
		Disp16[ER*m*]	2	2	-	-	-	1
		Disp6[BP]	2	2	-	-	-	1
		Disp6[FP]	2	2	-	-	-	1
		Dadr	2	2	-	-	-	1
	R*n*	[EA]	1	1	-	-	-	-
		[EA+]	1	1	-	-	-	-
		[ER*m*]	1	1 or 2	-	-	-	1
		Disp16[ER*m*]	2	2	-	-	-	1
		Disp6[BP]	2	2	-	-	-	1
		Disp6[FP]	2	2	-	-	-	1
		Dadr	2	2	-	-	-	1
	XR*n*	[EA]	2	2	-	-	-	-
		[EA+]	2	2	-	-	-	-
	QR*n*	[EA]	4	4	-	-	-	-
		[EA+]	4	4	-	-	-	-

Control Register Access Instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
ADD	SP	*#signed8*	1	1	-	-	-	-
MOV	ECSR	R*m*	1	1	-	-	-	-
	ELR	ER*m*	1	1	-	-	-	-
	EPSW	R*m*	1	1	-	-	-	-
	ER*n*	ELR	1	1	-	-	-	-
		SP	1	1	-	-	-	-
	PSW	R*m*	1	1	-	-	-	-
		#unsigned8	1	1	-	-	-	-
	R*n*	CR*m*	1	1	-	-	-	-
		ECSR	1	1	-	-	-	-
		EPSW	1	1	-	-	-	-
		PSW	1	1	-	-	-	-
	SP	ER*m*	1	1	-	-	-	-

PUSH/POP Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
PUSH	EA	1	1	-	-	-	1
	ELR	1 / 2 (*1)	1 / 2 (*1)	-	-	-	1
	EA,ELR	2 / 3 (*1)	2 / 3 (*1)	-	-	-	1
	EPSW	1	1	-	-	-	1
	EPSW,EA	2	2	-	-	-	1
	EPSW,ELR	2 / 3 (*1)	2 / 3 (*1)	-	-	-	1
	EPSW,ELR, EA	3 / 4 (*1)	3 / 4 (*1)	-	-	-	1
	LR	1 / 2 (*1)	1 / 2 (*1)	-	-	-	1
	LR,EA	2 / 3 (*1)	2 / 3 (*1)	-	-	-	1
	LR,ELR	2 / 4 (*1)	2 / 4 (*1)	-	-	-	1
	LR,EA,ELR	3 / 5 (*1)	3 / 5 (*1)	-	-	-	1
	LR,EPSW	2 / 3 (*1)	2 / 3 (*1)	-	-	-	1
	LR,EPSW,EA	3 / 4 (*1)	3 / 4 (*1)	-	-	-	1
	LR,EPSW,ELR	3 / 5 (*1)	3 / 5 (*1)	-	-	-	1
	LR,ELR,EPSW,EA	4 / 6 (*1)	4 / 6 (*1)	-	-	-	1
	ER*n*	1	1	-	-	-	1
	QR*n*	4	4	-	-	-	1
	R*n*	1	1	-	-	-	1
	XR*n*	2	2	-	-	-	1
POP	EA	2	2	-	-	-	1
	EA,LR	3 / 4 (*1)	3 / 4 (*1)	-	-	-	1
	EA,PC	5 / 6 (*1)	10 / 11(*1)	-	-	-	1
	EA,PC,LR	6 / 8 (*1)	11 / 13 (*1)	-	-	-	1
	EA,PC,PSW	6 / 7 (*1)	11 / 13 (*1)	-	-	-	1
	EA,PC,PSW,LR	7 / 9 (*1)	12 / 14 (*1)	-	-	-	1
	EA,PSW	3	3	-	-	-	1
	EA,PSW,LR	4 / 5 (*1)	10 / 12 (*1)	-	-	-	1
	LR	1 / 2 (*1)	1 / 2 (*1)	-	-	-	1
	LR,PSW	2 / 3 (*1)	2 / 3 (*1)	-	-	-	1
	PC	3 / 4 (*1)	8 / 9 (*1)	-	-	-	1
	PC,LR	4 / 6 (*1)	9 / 11(*1)	-	-	-	1
	PC,PSW	4 / 5 (*1)	9 / 10 (*1)	-	-	-	1
	PC,PSW,LR	5 / 7 (*1)	10 / 12 (*1)	-	-	-	1
	PSW	1	1	-	-	-	1
	ER*n*	1	1	-	-	-	1
	QR*n*	4	4	-	-	-	1
	R*n*	1	1	-	-	-	1
	XR*n*	2	2	-	-	-	1

(*1) The execution cycle in memory mode is SMALL or LARGE. (SMALL model/LARGE model)

Coprocessor Data Transfer Instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
MOV	CR*n*	R*m*	1	1	-	-	-	-
	CER*n*	[EA]	1	1	1	5	1	1
		[EA+]	1	1	1	5	1	1
	CQR*n*	[EA]	4	4	4	15	1	1
		[EA+]	4	4	4	15	1	1
	CR*n*	[EA]	1	1	1	5	1	1
		[EA+]	1	1	1	5	1	1
	CXR*n*	[EA]	2	2	2	10	1	1
		[EA+]	2	2	2	10	1	1
MOV	R*n*	CR*m*	1	1	-	-	-	-
	[EA]	CER*m*	1	1	1	5	1	1
	[EA+]	CER*m*	1	1	1	5	1	1
	[EA]	CQR*m*	4	4	4	15	1	1
	[EA+]	CQR*m*	4	4	4	15	1	1
	[EA]	CR*m*	1	1	1	5	1	1
	[EA+]	CR*m*	1	1	1	5	1	1
	[EA]	CXR*m*	2	2	2	10	1	1
	[EA+]	CXR*m*	2	2	2	10	1	1

EA Register Data Transfer Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
LEA	[ER*m*]	1	1	-	-	-	-
	Disp16[ER*m*]	2	2	-	-	-	-
	Dadr	2	2	-	-	-	-

ALU Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
DAA	R*n*	1	1	-	-	-	-
DAS	R*n*	1	1	-	-	-	-
NEG	R*n*	1	1	-	-	-	-

Bit Access Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
SB	*Dbitadr*	2	3	-	-	1	-
	R*n*.*bit_offset*	1	1	-	-	-	-
RB	*Dbitadr*	2	3	-	-	1	-
	R*n*.*bit_offset*	1	1	-	-	-	-
TB	*Dbitadr*	2	3	1	5	1	-
	R*n*.*bit_offset*	1	1	-	-	-	-

PSW Access Instructions

Instruction	Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
	No wait mode	Wait mode	No wait mode	Wait mode		
EI	1	1	-	-	-	-
DI	3	3	-	-	-	-
SC	1	1	-	-	-	-
RC	1	1	-	-	-	-
CPLC	1	1	-	-	-	-

Sign Extension Instruction

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
EXTBW	ER*n*	1	1	-	-	-	-

Branch Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
B	*Cadr*	2	6	-	-	-	1
	ER*n*	2	6	-	-	-	1
BL	*Cadr*	2	6	-	-	-	1
	ER*n*	2	6 or 7	-	-	-	1

Conditional Relative Branch Instructions

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
BGE	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BLT	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BGT	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BLE	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BGES	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BLTS	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BGTS	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BLES	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BNE	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BEQ	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BNV	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BOV	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BPS	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BNS	*Radr*	1 or 2(*1)	1 or 7(*1)	-	-	-	1
BAL	*Radr*	2	7	-	-	-	1

(*1) The larger number is for when the branching condition is satisfied; the smaller number is for when the branching condition is not satisfied.

Multiplication and Division Instructions

Instruction			Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
			No wait mode	Wait mode	No wait mode	Wait mode		
MUL	ER*n*	R*m*	9	9	-	-	-	-
DIV	ER*n*	R*m*	17	17	-	-	-	-

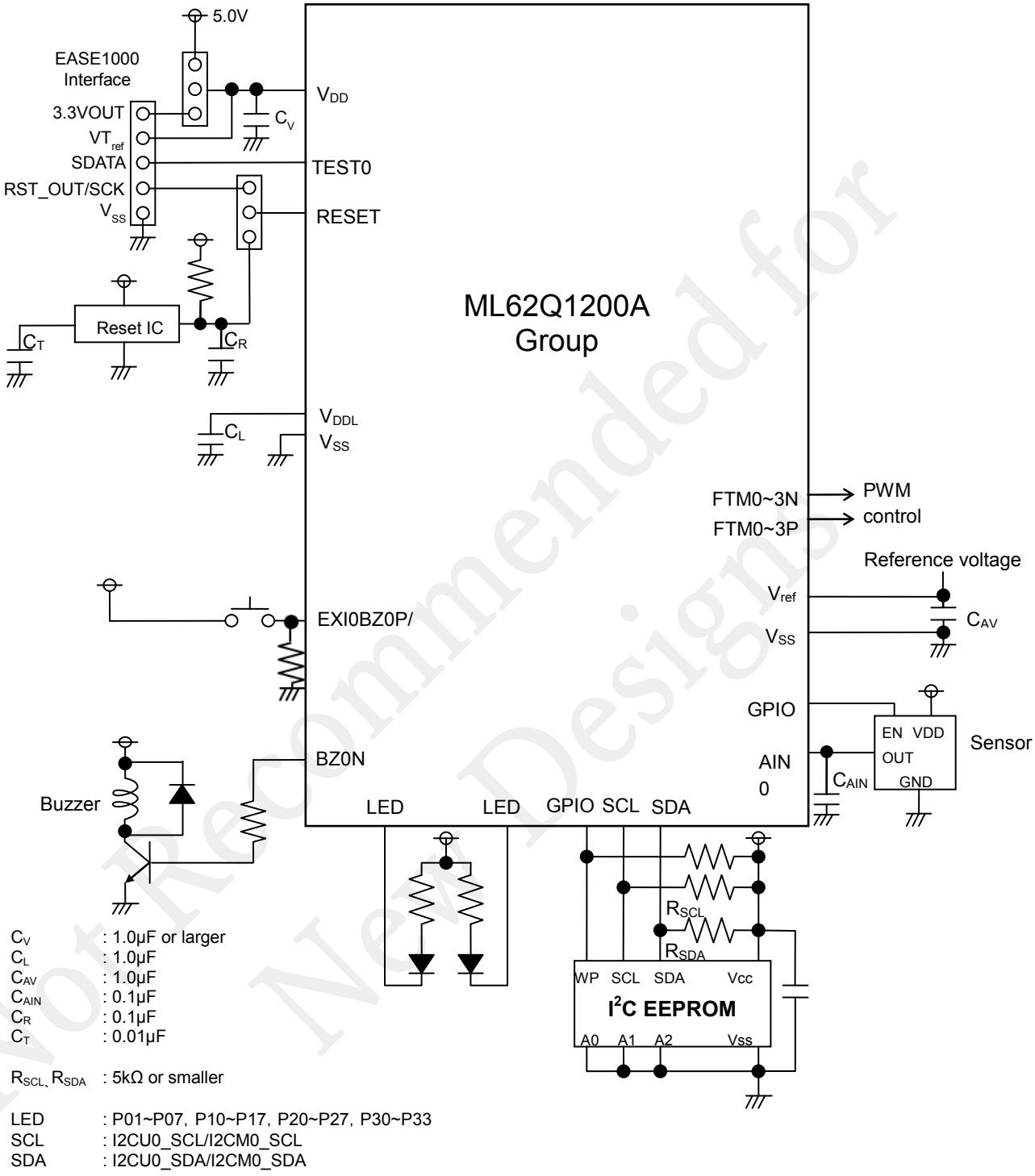
Interrupts

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
SWI	#*snum*	3	10	-	-	-	1
BRK		7	18	-	-	-	1
Interrupt transfer cycle		3	10	-	-	-	1

Miscellaneous

Instruction		Min. execution cycle		ROM reference cycle		Effect of DSR access	Effect of [EA+] addressing
		No wait mode	Wait mode	No wait mode	Wait mode		
NOP		1	1	-	-	-	-
DEC	[EA]	2	2	-	-	1	1
INC	[EA]	2	2	-	-	1	1
RT		2	6	-	-	1	1
RTI		2	6	-	-	1	1

Appendix D. Application Circuit Example



[Note]
Place the capacitor for V_{DDL} pin as close to the LSI power pins as possible.

Appendix E. Summary 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 up chapter by chapter when coding the program or evaluating it.

Common to all Chapters

[ ] For registers with word symbol, word access is possible. For word access, specify an even address. See "List of Registers" in each chapter.

Chapter 1 Overview

(Section 1.3.4 "TERMINATION OF UNUSED PINS".)

[ ] For unused input ports or unused input/output ports, if an unstable middle level voltage is supplied to the corresponding pins which are configured as inputs without pull-up register or input/output mode, shoot-through may become excessively large. Therefore, it is recommended to configure those pins as either input mode with a pull-up resistor or output mode.

Chapter 2 CPU and Memory Space

(Section 2.4 "Program Memory Space".)

[ ] The code option area(64 bytes) is not available for program code area. For details of the code option area, see Chapter 26 "Code Option" and make sure setting data.

[ ] It is recommended for failsafe to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space by using HTU8. See "HTU8 User's Manual" and "nX-U16/100 Core Instruction Manual".

[ ] Do not access unused area to avoid the CPU runs in correct.

(Section 2.5 "Data Memory Space".)

[ ] The contents of RAM area are undefined at power-up and system reset. Initialize the area by thesoftware.

[ ] The test area 1K byte(512 word) has MCU's unique data and the area cannot be used as the program memory.

Chapter 3 Reset Function

(Section 3.3.1 "Operation in Reset Mode".)

[ ] The system resets do not initialize data memory(RAM) and undefined SFRs. Please initialize them by the software.

[ ] The BRK instruction resets only CPU and does not reset peripheral circuit and other hardwares. Use RESET_N pin reset or WDT reset for surely initializing the LSI in case an unexpected error is detected.

[ ] The block reset control registers (BRECON0 to 3) reset only the peripheral circuits and the CPU and any other hardware are not initialized, also the LSI does not enter the system reset mode.

(Section 3.3.3 "RESET_N pin".)

[ ] Internal pull-up resistor is not installed. Have the pull-up register externally.

[ ] Remain the "L" level for the time of reset hold time (PRST) or longer.

(Section 3.3.4 "Power-On Reset".)

[ ] Rise the VDD up to 1.8V or higher at the power up.

[ ] At the power-on, the reset is released when the power voltage(VDD) is higher than the power-on reset detect voltage(VPOR) and the CPU starts running the program with low-speed clock(LSCLK/approx. 32.768kHz). Make one of following process when switching the CPU clock to a high-speed clock.

- Remain reset by the VLS function as long as the VDD is lower than the operate-able voltage (1.8V) for the high-speed clock.
- Check with the VLS function if the VDD is higher than operate-able voltage (1.8V) to switch the clock.

See For details about the VLS, see Chapter 22 " Voltage Level Supervisor (VLS)".

[ ] At the power-off, the reset occurs when the power voltage(VDD) is lower than the power-on reset detect voltage(VPOR), however make the VLS reset before the VDD is lower than the minimum operating voltage described in the electrical characteristics of data sheet. Also, confirm if the voltage has returned within the operating voltage when the CPU restarts running the program.

Chapter 4 Power Management

(Section 4.2.2 "Stop Code Acceptor (STPACP)".)

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

(Section 4.2.3 "Standby Control Register (SBYCON)".)

[ ] The operating state does not enter the standby mode under the condition that both an interrupt enable flag and an interrupt request flag are "1" that is requesting the interrupt to the CPU.

[ ] When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condtion of MIE flag of the program status word (PSW) is "0", it cancels the stnadby mode only and does not enter the interrupt processing. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".

[ ] Insert two NOP instructions after the instruction of that sets HLT, STP, HLTH and STPD bit to "1". The operation without the two NOP instructions is not guaranteed. When using the automatic CRC calculation mode, See the program example described in Chapter 19.3.3. "Automatic CRC Calculation Mode".

[ ] If two bits or more in the SBYCON are set to "1" at the same time, the setting are gets invalid and continues the program rum mode.

(Section 4.2.4 – Section 4.2.11.)

[ ] To enable operation of the peripheral circuits, cancel the reset by the block reset control regiser (BRECONn) after enabling the clock supply by the block clock control register (BCKCONn).

(Section 4.2.6 "Block Clock Control Register 2 (BCKCON2)".)

[ ] DCKACC can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified in the target option of the integrated development environment IDEU8.

(Section 4.3.6 "Note on Return Operation from the standby modes".)

[ ] The operation of returning from the STOP/STOP-D mode and the HALT/ HALT-H mode varies according to the interrupt level (ELEVEL) of the program status word (PSW), master interrupt enable flag (MIE), and the contents of the interrupt enable register (IE0 to IE3), as well as whether the interrupt is non-maskable or maskable.

[ ] Insert two NOP instructions after the standby mode (HALT, STOP, HALT-H, STOP-D mode) transition instructions, because the maximum two instructions will be executed after releasing the standby mode and before transferring to the interrupt. When the master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", instructions will be executed after the two NOP execution and the interrupt transferring cycles. When the MIE is "0" the CPU executes the two NOP instructions and continues executing the next instructions without transferring to the interrupt.

(Section 4.3.7 "Operation of Functions in the standby modes".)

[ ] To operate the peripheral circuits in the HALT-H mode, select the low-speed clock for the operation clock of the peripheral circuits.

[ ] If the system clock is switched to the high-speed clcok after the STOP or STOP-D mode is released before the wake-up time is passed, the CPU is hold to run the program because the clock has not been supplied.

[ ] When the high-speed clock wake-up time setting register (FHWUPT) is "0x00", the output clock of PLL oscillation is masked for approx.2.5ms. The HSCLK starts to be supplying after waiting the 2.5ms. The SYSTEM clock is also stopped during the 2.5ms.

[ ] When the high-speed clock wake-up time setting register (FHWUPT) is "0x01", the frequency of PLL oscillation clock gradually increses from approx. 1MHz and will get the correct frequency (16MHz/24MHz/32MHz) set by the code option withing the approx.2.5ms. The accuracy of the frequency described in the data sheet cannot be guranteed during the 2.5ms, however it can be used for the system clock without waiting the 2.5ms.

(Section 4.3.8 "Block Control Function".)

[ ] If the clock supply is stopped to the peripheral circuits without reseting them by the BRECONn(n=0 to 3), it is possible that output levels of timer, serial communications and buzzer are fixed and it makes a large current. Also, it is possible the sucessive A/D conveter, D/A converter and analog comparator may stop on the condition of the large current is flowing. Thefore, please make sure to stop the clock after reseting the periperals by using the block reset control register BRECONn(n=0 to 3) register.

Chapter 5 Interrupts

(Section 5.2.10 "Interrupt Level Control Enable Register (ILEN)".)

[ ] Disable the inerrupt level control function by reseting the ILE bit to "0" after resetting the Interrupt level control register ILCn0 and ILCn1(n=0~7) to "0x00" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled(IE0~IE7 registers are "0x00").

[ ] Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0~IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.

(Section 5.3 "Description of Operation".)

[ ] For failsafe, define also unused interrupt vectors. If the unused interrupt happened it signifys possibility of that the CPU went out of control. Therefore, it is recommended to make the Watch Dog Timer overflow and reset by program codes execute an infinite loop.

(Section 5.3.4 "Notes on Interrupt Routine (when the Interrupt Level Control is disabled)".)

[ ] When "0" is written to the interrupt level control enable register (ILEN), the interrupt level control is disabled.

(Section 5.3.5 "Flow chart of interrupt processing when the Interrupt Level Control is enabled".)

[ ] For considering the processing of non-maskable interrupt, refer to the flow charge in the case of the multiple interrupts are enabled. ELR2 and EPSW2 will be saved to the stack in that case.

[ ] When programming by the C language, the program code for the save/reload are generated by the Compiler (Push/Pop instruction). Other codes for enabling/disabling the interrupt by EI/DI instruction and writing to the current interrupt request level management register(CIL).

(Section 5.3.6.1 "When programming interrupts defined as it disables other multiple interrupts".)

[ ] Do not enable any interrupt in a function called from an interrupt function defined as it does not allow the multiple interrupts. Otherwise, the program may run out of control when the multiple interrupts occur.

Chapter 6 Clock Generation Circuit

(Section 6.1.2 "Configuration".)

[ ] After power-on or system reset, the low-speed RC oscillation clock (approx. 32.768 kHz) is initially selected and supplied to the system clock (SYSTEMCLK).

(Section 6.2.2 "High-Speed Clock Mode Register (FHCKMOD)".)

[ ] When the voltage of V_{DD} is $1.6V \leqq V_{DD} < 1.8V$, set the system clock to 4 MHz or lower. If it exceeds 4 MHz, the operation is not guaranteed.

[ ] For output of the high-speed clock (OUTHSCLK), the output clock frequency is limited according to the voltage of VDD.

$1.6V \leqq VDD < 1.8V$: Select 4 MHz or lower

$1.8V \leqq VDD \leqq 5.5V$: Select 12 MHz or lower

(Section 6.2.5 "Low-Speed RC Oscillation Frequency Adjustment Register (LRCADJ)".)

[ ] Use the RC oscillation adjustment sample software provided by LAPIS. Otherwise, the operation is not guaranteed.

(Section 6.3.4 "Switching of System Clock".)

[ ] While the system clock is operating at the low speed, if interrupts of the peripheral circuits are enabled for high-speed clock operation, the interrupt processing of the CPU may not be in time. Consider the timing of the interrupt cycles and the operating frequency of the CPU.

Chapter 7 Time Base Counter

(Section 7.2.2 "Low-Speed RC Oscillation Frequency Adjustment Register (LRCADJ)".)

[ ] A time base counter interrupt may occur depending on a write timing to the LTBR. See the program example for initializing described in "7.3.1 Operation of the Low-speed Time Base Counter".

[ ] T128HZ ~ T1HZ signals have "0" level in the first half cycle and "1" level in the last half. For example, T1HZ signal gets reset to "0" by writing any data to LTBR and it get to "1" about 0.5sec later and returns to "1" about 1sec later from the reset. The low-speed time base counter interrupt occurs at the falling edge ("1" to "0") of the signal. See "Time base counter interrupt timing and reset timing of reset by writing to LTBR" for details about the waveform of signal.

(Section 7.2.4 "Low Speed Time Base Counter Interrupt Selection Register (LTBINT)".

[ ] A time base counter interrupt may occur depending on a write timing to the LTBINTL or LTLBINTH. See the program example for initializing described in "7.3.1 Operation of the Low-speed Time Base Counter".

(Section 7.3.1 "Operation of Low Speed Time Base Counter".

[ ] The time from when writing the LTBR register until when the 1st low-speed counter interrupt request generates, is not guranteed. Measure times with reference to the interval of interrupt occurrence.

Chapter 8 16bit Timer

(Section 8.1.2 "Configuration".)

[ ] When a 16-bit timer is used as two channels of 8-bit timer, the same clock settings and interrupts are applied.

[ ] For TMHnOUT (timer out) of the 8-bit timer mode, values on the upper side ("TMHnDH" and "TMHnCH") are output.

(Section 8.2.2 "16-Bit Timer n Data Register (TMHnD: n = 0 to 5)".)

[ ] Set TMHnD when the 16-bit timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are "0"). When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD. When "0x00" is written in TMHnDL/TMHnDH in the 8-bit timer mode, "0x01" is set in TMHnDL/TMHnDH.

(Section 8.2.4 "16-Bit Timer n Mode Register (TMHnMOD: n = 0 to 5)".)

[ ] Input the pulse for the external trigger with the width of two timer clocks or longer.

[ ] Set TMHnMOD when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are "0"). If it is changed while it is operating, the operation is not guaranteed. In the 8-bit timer mode, the operation mode specified by THnCS0 to 1, THnDIV2 to 0 and THnOST are common for both two channels.

(Section 8.2.7 "16-Bit Timer Start Register (TMHSTR)".)

[ ] Set TMHSTR when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are "0").

(Section 8.3.3 "External Input Count Timing".)

[ ] The external input pluse with width shorter than two clocks of timer clock may be ignored. Input the external clock signal with the two clocks width of timer clock or longer.

[ ] The external input signal for the 16bit Timer (EXTRGn) comes through the sampling controller in the external interrupt function. The sampling for the external interrupt is selectable to use or not. See Chapter 18 "External Interrupt Control Circuit".

Chapter 9 Functional Timer

(Section 9.2.11 "FTMn Trigger Register 1 (FTnTRG1: n = 0, 1, 2, 3)".)

[ ] Enable the trigger event after setting the noise filter function. Othewise, the trigger may occur immediately after setting this register.

(Section 9.2.17 "FTM Common Start Register (FTCSTR)".)

[ ] Set the FTCSTR register when the FTMn stops (when the FTnSTA bit of FTnSTAT register is "0").

(Section 9.2.18 "FTM Common Stop Register (FTCSTP)".)

[ ] Set the FTCSTP register when the FTMn is running (when the FTnSTA bit of FTnSTAT register is "1").

Chapter 10 Watchdog Timer

(Section 10.1.1 "Features".)

[ ] The watchdog timer is for monitoring the CPU's out of control, does not guarantee the function as a general timer.

(Section 10.2.2 "Watchdog Timer Control Register (WDTCON)".)

[ ] When the interrupt level (ELEVEL) of the U16 core is 2 or higher (during non-maskable interrupt or emulator interrput), the WDT counter cannot be cleared.

(Section 10.2.3 "Watchdog Timer Mode Register (WDTMOD)".)

[ ] When using the counter clear enable period is 50% or 75% of overflow period, the WDT interrupt does not occur. The first overflow generates the WDT reset.

(Section 10.2.5 "Watchdog Status Register (WDTSTA)".)

[ ] Max.two clock of WDTCLK(approx.1KHz) is required between the timing of writing "0x5A" and "0xA5" to the WDTCON and the timing of WDT counter is cleared.

[ ] Confirm if WDTCLR1 is "0" before entering to STOP mode or STOP-D mode.

[ ] Confrim if both WDTCLR1 and WDTCLR2 are "0" just after the WDT clear process, before changin the data of Watch Dog Timer Mode reigster(WDTMOD).

(Section 10.3 "Description of Operation".)

[ ] The watchdog timer cannot detect all the abnormal operations. Even if the CPU loses control, the watchdog timer cannot detect the abnormality in the operation state in which the WDT counter is cleared. It is recommended to clear the WDT counter in a loop of main routine for the fail-safe.

[ ] It's better to use the RC1KHZ for the WDT conter clock for the fail-safe because the clock independent from the sytem clock, however the RC1KHZ is less accurate than LSCLK. Use the LSCLK if high accurate clock is required.

[ ] The watchdog timer stops in the STOP/STOP-D mode.

(Section 10.3 "Operation when WDT clear enable time period is 75% or 50% of overflow cycle".)

[ ] When selecting the 75% or 50% of overflow period as the WDT clear enable time period, the WDT interrupt does not occur, however define the WDT interrupt service function in the program source code. For fale-safe, it is recommended to make program codes that forcely generate the WDT invalid clear reset.

[ ] When selecting the configuration of the overflow cycle is 125ms or longer, use the low-speed RC oscillation clcok for the WDT count clock by setting the Code Option. The WDT dedicated clock RC1KZ can not be used as the frequency is not accurate.

[ ] As the interrupt level (ELEVEL) of the Program Status Word(PSW) in the CPU is set to "2" in the WDT interrupt service routine, the WDT counter cannot be cleared. Clear the WDT counter when the ELEVEL is "0" or "1".

[ ] It is recommended to clear the WDT counter in a loop of main routine for the fail-safe.

Chapter 11 Serial Communication Unit

(Section 11.2.2 "Serial Communication Unit n Transmit/Receive Buffer (SDnBUF)".)

[ ] In the half-duplex communication mode of UART, be sure to select the transmit mode by setting Un0IO and Un1IO bit of the UARTn mode register (UAn0MOD, UAn1MOD) before writing the transmit data in SDnBUFL and SDnBUFH.

[ ] Do not perform write-operation to the SDnBUF in the SSIO slave receive mode.

(Section 11.2.3 "Serial Communication Unit n Mode Register (SUnMOD)".)

[ ] Be sure to set the SUnMOD register while communication is stopped (SUnCON=0x00). Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

(Section 11.2.4 "Serial Communication Unit n Transmission Interval Setting Register (SUnDLY)".)

[ ] Set "0x00" to the SUnDLYL register in the SSIO slave mode.

[ ] The SUnDLYL register is disabled in the SSIO master receive mode.

(Section 11.2.6 "Synchronous Serial Port n Mode Register (SIOnMOD)".)
[ ] Be sure to set the SIO0MOD register while communication is stopped. Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.
[ ] Set the S0CK4 to S0CK0 bits to 4MHz or below.
[ ] Enable the high-speed oscillation when selecting the slave mode. See Chapter 6 "Clock Generation Circuit" for details on how to enable the high-speed oscillation.
[ ] The maximum frequency of communication clock is 1MHz in the slave mode.
[ ] Specify the frequency of transfer clock as 24MHz or lower.

(Section 11.2.8 "UARTn0 Mode Register (UAn0MOD)".)
[ ] Be sure to set the UAn0MOD register while communication is stopped. Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

(Section 11.2.11 "UARTn0 Mode Register (UAn0MOD)".)
[ ] Be sure to set the UAn0BRT and UAn1BRT register while communication is stopped. Do not rewrite it during communication.

(Section 11.2.14 "UARTn0 Status Register (UAn0STAT)".)
[ ] The Un0OER bit becomes "1" if the previous receive data is not read even after reception is stopped by the Un0EN bit and restarted. Therefore, set the Un0EN bit to "1" after reading the SDnBUFL, or when reception is completed, be sure to read the SDnBUFL even if the data is not necessary.
[ ] Do not write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by using the bit symbol. Write them by the byte-access.

(Section 11.2.15 "UARTn1 Status Register (UAn1STAT)".)
[ ] The Un1OER bit becomes "1" if the previous receive data is not read even after reception is stopped by the Un1EN bit and restarted. Therefore, set the Un1EN bit to "1" after reading the SDnBUFH, or when reception is completed, be sure to read the SDnBUFH even if the data is not necessary.
[ ] Do not write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by using the bit symbol. Write them by the byte-access.

(Section 11.3.2.8 "Receive Margin".)
[ ] When designing the system, adjust the baud rate in the UAn0BRT, UAn1BRT, UAn0BRC, and UAn1BRC registers, considering difference of the baud rate between the transmit side and receive side, and also the delay of start bit detection.

Chapter 12 I^2C Bus Unit
(Section 12.2.2 "I^2C Bus Unit 0 Mode Register (I2U0MSS)".)
[ ] Do not write to SFRs for slave function in the master mode and do not write SFRs for master function in the slave mode.
[ ] When using the master function, do not connect multiple master devices on the I2C bus.
[ ] When using the salve function, do not connect multiple slave devices on the I2C bus.
[ ] When using the salve function, switch the system clock to the high-speed clock if releasing the communication wait status.

(Section 12.2.6 "I^2C Bus 0 Control Register (Master) (I2UM0CON)".)
[ ] Do not update the I2UM0ACT bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.
[ ] When the I2UM0ST bit is "1", write the I2UM0CON register in the control register setting wait state.

(Section 12.2.7 "I^2C Bus 0 Mode Register (Master) (I2UM0MOD)".)
[ ] Specify the I^2C operation clock as follows.

When the HSCLK is 32MHz

Standard mode	: I^2C operation clock HSCLK~1/4HSCLK
Fast mode	: I^2C operation clock HSCLK, 1/2HSCLK
1Mbps mode	: I^2C operation clock HSCLK, 1/2HSCLK

When the HSCLK is 24MHz

Standard mode	: I^2C operation clock HSCLK~1/4HSCLK
Fast mode	: I^2C operation clock HSCLK, 1/2HSCLK
1Mbps mode	: I^2C operation clock HSCLK, 1/2HSCLK

When the HSCLK is 16MHz
Standard mode : I2C operation clock HSCLK, 1/2HSCLK
Fast mode : I2C operation clock HSCLK
1Mbps mode : I2C operation clock HSCLK

(Section 12.2.8 "I2C Bus 0 Status Register (Master) (I2UM0STA)".)
[ ] Do not update each bit of the I2UM0STA bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.

(Section 12.2.12 "I2C Bus 0 Control Register (Slave) (I2US0CON)".)
[ ] Switch the system clock to the high-speed clock if releasing the communication wait status.
[ ] Do not update the I2US0ACT bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.

(Section 12.2.13 "I2C Bus 0 Mode Register (Slave) (I2US0MD)".)
[ ] Set I2US0EN bit to "0" to stop the operation before entering STOP/STOP-D mode.

(Section 12.2.14 "I2C Bus 0 Status Register (Slave) (I2US0STA)".)
[ ] Do not update each bit of the I2UM0STA register by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.

(Section 12.3.6 "Pin Settings".)
[ ] Use external pull-up resistors for SDA pin and SCL pin refering to the I2C bus specification. The internal pull-up resistors can not satisfy the I2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.

Chapter 13 I2C Bus Master
(Section 13.2.5 "I2C Master 0 Control Register (I2M0CON)".)
[ ] Do not update the I2MnACT bit by using the bit symbol. Update it by using a byte access, not so that unintented bits are changed by the bit access instructions.
[ ] When the I2MnST bit is "1", write the I2MnCON register in the control register setting wait state.

(Section 13.2.6 "I2C Master 0 Mode Register (I2M0MOD)".)
[ ] Specify the I2C operation clock as follows.
When the HSCLK is 32MHz
Standard mode : I2C operation clock HSCLK~1/4HSCLK
Fast mode : I2C operation clock HSCLK, 1/2HSCLK
1Mbps mode : I2C operation clock HSCLK, 1/2HSCLK
When the HSCLK is 24MHz
Standard mode : I2C operation clock HSCLK~1/4HSCLK
Fast mode : I2C operation clock HSCLK, 1/2HSCLK
1Mbps mode : I2C operation clock HSCLK, 1/2HSCLK
When the HSCLK is 16MHz
Standard mode : I2C operation clock HSCLK, 1/2HSCLK
Fast mode : I2C operation clock HSCLK
1Mbps mode : I2C operation clock HSCLK

(Section 13.3.4 "Pin Settings".)
[ ] Use external pull-up resistors for SDA pin and SCL pin refering to the I2C bus specification. The internal pull-up resistors can not satisfy the I2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.

Chapter 14 DMA Controller

(Section 14.2.2 "DMA Channel n Mode Register (DCnMOD: n = 0, 1)")

[ ] Set the bits except for DCnSTRG bit when the transfer is disabled (DCnEN = 0).

[ ] When selecting the 16bit timer DMA request, select the 16bit timer mode by setting THn8BM bit of 16bit timer n mode register (TMHnMOD) to "0".

(Section 14.2.3 "DMA Channel n Transfer Count Register (DCnTN: n = 0, 1)")

[ ] Set the DCnTN when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).

[ ] If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnTN is not guaranteed. Reconfigure the DCnTN when restarting the transfer.

(Section 14.2.4 "DMA Channel n Transfer Source Address Register (DCnSA: n = 0, 1)")

[ ] Set the DCnTN when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).

[ ] If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnSA is not guaranteed. Reconfigure the DCnSA when restarting the transfer.

(Section 14.2.5 "DMA Channel n Transfer Destination Address Register (DCnDA: n = 0, 1)")

[ ] Set the DCnTN when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).

[ ] If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the value of the DCnDA is not guaranteed. Reconfigure the DCnDA when restarting the transfer.

(Section 14.2.6 "DMA Transfer Enable Register (DCEN)")

[ ] Set the DCnTN when the transfer is disabled (DCnEN = 0). The DCnTN is not writable if the transfer is enabled (DCnEN = 1).

[ ] When the specified transfer count is completed, the channel corresponding bit of the DMA interrupt status register (DSTATL) is set to "1". Be sure to clear the status bit before performing the next DMA transfer. When the status is set to "1", channel operation cannot be enabled. Clear the DCnISTA regardless using or not using the DMA interrupt.

(Section 14.3.1 "DMA Operation Procedure")

[ ] The DMA transfer is held if the CPU continuously access the data memory, because the data memory access has a higher priority than the DMA transfer. If an interval is short, the transfer trigger may be overwritten. Take the transfer trigger interval time longer than that of the CPU continues the data memory access. Take Take four system clocks or longer in the transfer intervals in the case there is no data memory access.

[ ] If a transfer trigger and a software trigger are generated at the same time, the transfer trigger is overwritten. Pay attention to a timing when a software trigger is generated.

[ ] The DMA transfer is not available in HALT mode, HATL-H mode, STOP mode and STOP-D mode.

[ ] Stop the clock for DMA by setting DCKDMA bit of BCKCON2 register, before entering into the HALT mode, HATL-H mode, STOP mode and STOP-D mode. Any trigger occurred in the standby mode is ignored. If restart to supply the clock by setting the DCKDMA bit to"0" the DMA transfer restarts. SFRs related to the DMA is not accessible when the clock for the DMA is stopped. Supply the clock before accessing the SFRs.

[ ] Specify an area where the RAM and SFR exist for the transfer source/destination address. If a non-existing are is specified, "0" is transferred.

[ ] Use the DMA transfer when the CPU mode is the wait mode.

Chapter 15 Buzzer

(Section 15.3.2.1 "Buzzer Output Waveform")

[ ] When selecting the intermittent sound 1 mode, intermittent sound 2 mode or single sound mode, the buzzer output is not started from the first waveform dpending on the timing of BZ0RUN bit is set to "1". To prevent it, set the BZ0RUN bit syncronizing at the falling edge of T8HZ or T1HZ signal by using the T8HZ/T1HZ time base counter interrupt or checking T8HZ/T1HZ bit of LTBR register.

Chapter 17 GPIO port

(Section 17.2.3 "Port n Mode Register 01 (PnMOD01) n: Port Number (0~3)")

[ ] Set the port n mode register before setting the external interrupt registers (EICON0 and EIMOD0) and the interrupt enable register (IE1), If setting the port n mode register when the interrupt is enabled, unexpected may happen.

(Section 17.3.8 "Notes for using the P00/TEST0 pin")

[ ] P00/TEST0 pin is used for the general port, the on-chip debug function or ISP funcction. Confirm the notes in each usage.

[ ] P00/TEST0 is initially configured as the input with pull-up register. If input "L" level at the initial setting, the input current flows.

Chapter 18 External Interrupt Control Circuit

(Section 18.2.3 "External Interrupt Mode Register 0 (EIMOD0)")

[ ] In the STOP/STOP-D/HALT-H mode, no sampling is performed regardless of the values set in PI7SM to PI0SM since the sampling clock stops. In this switching from "with sampling" to "without sampling", there is a time period (*) in which interrupts gets disabled. When entering to STOP/STOP-D/HALT-H mode, specify the external interrupt as "without sampling". After returning from STOP/STOP-D/HALT-H mode, specify the PI7SM to PI0SM as "with sampling" if needed.

(*) When entering to STOP/STOP-D mode: Max.30us
When returing from STOP/STOP-D mode: Max.250us(When LSCLK is selected for CPU),
Max.2.5ms(When HSCLK is selected for CPU)
When returing from HALT-H mode: Max.2.5ms(When HSCLK is selected for CPU)

[ ] If HSCLK is selected for the sampling block when the high-speed clock is not oscillating, the sampling circuit does not operate. If sampling is performed with HSCLK, enable the high-speed clock oscillation in advance. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".

Chapter 19 CRC Generator

(Section 19.2.2 "Automatic CRC Calculation Start Address Setting Register (CRCSAD)")

[ ] CRCSAD must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is invalid when the CRCAEN bit is "1".

[ ] Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bits 1 and bit0 are ignored, fixed to "0" internally during the calculation.

(Section 19.2.3 "Automatic CRC Calculation End Address Setting Register (CRCEAD)")

[ ] CRCEAD must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is invalid when the CRCAEN bit is "1".

[ ] Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bits 1 and bit0 are ignored, fixed to "0" internally during the calculation.

(Section 19.2.4 "Automatic CRC Calculation Start Segment Setting Register (CRCSSEG)")

[ ] CRCSSEG must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

(Section 19.2.6 "CRC Data Register (CRCDATA)")

[ ] CRCDATA must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

(Section 19.2.7 "CRC Calculation Result Register (CRCRES)")

[ ] CRCRES must be written when the CRCAEN bit of the CRC mode register (CRCMOD) is "0". Any write is disabled when the CRCAEN bit is "1".

(Section 19.3.1 "CRC Calculation Mode")

[ ] To perform CRC calculation when automatic CRC calculation is not completed, save the CRCRES value before calculation. After CRC calculation, move the saved value back to CRCRES and set CRCAEN to "1". When the LSI switches to the HALT mode, automatic CRC calculation can be restarted. The end addresses of the program code area are stored in CRCSAD and CRCSSEG. If the value is overwritten when CRCAEN is "0", data cannot be checked correctly.

(Section 19.3.3 "Automatic CRC Calculation Mode")

[ ] When transferring to the HALT/HALT-H mode in the automatic CRC calculation mode, insert following program codes after setting the HLT/HALTH bit, otherwise the operation can not be guaranteed.

```
HLT=1;    // or HLTH=1;
__asm("NOP");
__asm("DW 0CE08H");
__asm("DW 8 DUP 0FE8FH");
```

Chapter 20 Analog Comparator

(Section 20.1.3 "List of Pins")

[ ] When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to "0" as "Disable the input" and "Disable the output", othewise a shoot-through current may flow.

(Section 20.2.2 "Comparator 0 Control Register (CMP0CON)")

[ ] An infuluence of the noise is reduceable by preventing the switching of neighboring pins while reading the COMP0D bit when the comparator enables.

(Section 20.2.3 "Comparator 0 Mode Register (CMP0MOD)")

[ ] In the STOP/STOP-D/HALT-H mode, no sampling is performed regardless of the value set in CMP0CS1 and CMP0CS0 since the sampling clock stops. In this switching from "with sampling" to "without sampling", there is a time period (*) in which interrupts gets disabled. When entering to STOP/STOP-D/HALT-H mode, specify the external interrupt as "without sampling". After returning from STOP/STOP-D/HALT-H mode, specify the CMP0CS1 and CMP0CS0 as "with sampling" if needed.
(*) When entering to STOP/STOP-D mode: Max.30us
When returing from STOP/STOP-D mode: Max.240us(When LSCLK is selected for CPU),
Max.2.5ms(When HSCLK is selected for CPU)
When returing from HALT-H mode: Max.2.5ms(When HSCLK is selected for CPU)

[ ] When transferring to the HALT/HALT-H mode in the automatic CRC calculation mode, insert following program codes after setting the HLT/HALTH bit, otherwise the operation can not be guaranteed.

Chapter 21 D/A Converter

(Section 21.1.3 "List of Pins")

[ ] When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to "0" as "Disable the input" and "Disable the output", othewise a shoot-through current may flow.

(Section 21.2.2 "D/A Converter Enable Register (DACEN)")

[ ] An infuluence of the noise is reduceable by preventing the switching of neighboring pins while the D/A conversion is enabled.

Chapter 22 Voltage Level Supervisor (VLS)

(Section 22.2.2 "Voltage Level Supervisor Control Register (VLS0CON)")

[ ] During the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode. When the VLS0RF bit is set to "0", the LSI cannot enter the STOP/STOP-D mode.

[ ] During the single mode, the LSI cannot enter the STOP/STOP-D mode. When the single mode operation is completed (make sure that the VLS0EN bit is set to "0"), enable the STOP/STOP-D mode.

[ ] The VLS0 interrupt is not available to wake up from the STOP/STOP-D mode. Enter the STOP/STOP-D mode after disabling the EVLS0 bit of interrupt enable register IE0.

[ ] Even if a reset (VLS0 reset, WDT reset, ROM unused area access reset, or RAM parity error reset) other than the POR and pin reset is output, VLS0 remains operating.

(Section 22.2.3 "Voltage Level Supervisor Mode Register (VLS0MOD)")

[ ] Select always "With sampling" in the Voltage Level Supervisor 0 Sampling Register (VLS0SMP) when enabling the VLS0 interrupt request function in the supervisor mode.

[ ] During the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode. When the VLS0RF bit is set to "0", the LSI cannot enter the STOP/STOP-D mode.

(Section 22.3.1 "Supervisor Mode")

[ ] When the VLS is being stabilized, the LSI cannot enter the STOP/STOP-D mode. After starting the supervisor mode, make sure that the VLS0RF bit is set to "1" before enabling the STOP/STOP-D mode.

[ ] When the VLS is stopped (VLS0EN = "0") during the low voltage detection state (VLS0F = "1"), a VLS0 interrupt occurs.

Chapter 23 Successive Approximation type A/D Converter

(Section 23.1.3 "Supervisor Mode")

[ ] When using the analog comparator, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 3, m: bit number 0 to 7) to "0" as "Disable the input" and "Disable the output", othewise a shoot-through current may flow.

[ ] During the A/D conversion, influence of noise can be reduced by avoiding of switching other pins or by performing the A/D conversion in the HALT mode.

(Section 23.2.4 "SA-ADC Upper/Lower Limit Status Register 0 (SADULS0)")

[ ] Do not use bit access instructions and use word or byte access instructions for reading or writing the SADULS0 register.

[ ] When performing the A/D conversion only one time, confirm the bit of SAULS07~SAULS00 is "0" before setting SARUN bit to "1".

[ ] When performing the consecutive A/D conversion (SALPEN=1), confirm the bit of SAULS07~SAULS00 is "0", before the next A/D conversion ends.

(Section 23.2.5 "SA-ADC Upper/Lower Limit Status Register 1 (SADULS1)")

[ ] Do not use bit access instructions and use word or byte access instructions for reading or writing the SADULS1 register.

(Section 23.2.6 "SA-ADC Mode Register (SADMOD)")

[ ] Write "0" to SADMODH[7:1]. Operation when "1" is written to the bits cannot be guranteed.

(Section 23.2.7 "SA-ADC Control Register (SADCON)")

[ ] Start A/D conversion with one or more channels selected in the SA-ADC enable registers (SADEN0, SADEN1). If no channel is set, operation does not start.

[ ] Enter STOP/STOP-D mode after checking SARUN bit is "0", it does not enter the STOP/STOP-D mode when the SARUN bit is "1".

(Section 23.2.8 "SA-ADC Enable Register 0 (SADEN0)")

[ ] Do not start the A/D conversion when the all SACHn bits are "0". In that case the SARUN bit does not get to "1".

(Section 23.2.9 "SA-ADC Enable Register 1 (SADEN1)")

[ ] When using the channel 16 (SACH16), enable the intenal reference voltage/temperature sensor and select the internal reference voltage by setting the reference voltage control register (VREFCON).

(Section 23.2.11 "SA-ADC Upper/Lower Limit Mode Register (SADLMOD)")
[ ] The upper/lower limit detection function is available to make the interrupt request for the A/D conversion result on all seleted channels.

(Section 23.2.14 "SA-ADC Reference Voltage Control Register (VREFCON)")
[ ] It takes 200us(Max.) until the internal reference voltage gets stable after setting VREFEN bit to "1". Start the A/D conversion after waiting the stabilization time.
[ ] The internal reference voltage(Approx. 1.55V) can be output to the general port(P23) by setting the VREFEN bit to"1" and setting 0x70 to P2MOD3 register. However in that case, it is possbible to get incorrect A/D conversion results as affected by external factors.
[ ] When using the external reference voltage input from V_{REF} pin(P23), set VREFP1 bit to "0" and VREFP0 bit to "1" and P2MOD3 register to 0x00.

(Section 23.2.15 "SA-ADC Interrupt Mode Register (SADIMOD)")
[ ] If SALEN bit of the SA-ADC upper/lower limit mode register (SADLMOD) is set to "1", the interrupt from the upper/lower limit detection function gets enabled and the setting for the SADIMD bit of SA-ADC Interrupt Mode Register (SADIMOD) is invalid.

(Section 23.2.16 "SA-ADC Trigger Register (SADTRG)")
[ ] When selecting the 16bit Timer interrupts (TM0INT and TM1INT), set the THn8BM bit of the 16bit Timer n Mode Register (TMHnMOD) to "0" to select the 16bit timer mode.

Chapter 24 Regulator
(Section 24.1.3 "List of Pins")
[ ] For increasing the noise resistance, place the bypass capacitor (Cv 1μF) and the capacitor for V_{DDL} close to the power pins and keep the traces from capacitor to the pins as short as possible without going via through holes.

Chapter 25 Flash Memory
(Section 25.2.4 "Flash Data Register 0 (FLASHD0)")
[ ] Specify a segment address to the FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.
[ ] Back Ground Operation(BGO) function allows CPU continue running the program codes while programming the data flash memory. Confirm the end of programming by checking FDPRSTA bit of Flash Status Register(FLASHSTA).
[ ] Erase data in the addresses in advance. Programmed data without erase cannot be guranteed.
[ ] Do not access to unused area, otherwise the CPU may work incorrectly.

(Section 25.2.5 "Flash Data Register 1 (FLASHD1)")
[ ] Specify a segment address to the FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.

(Section 25.2.8 "Flash Self Register (FLASHSLF)")
[ ] Reset the FSELF bit to "0" before reading the Flash memory.

(Section 25.2.9 "Flash Status Register (FLASHSTA)")
[ ] Perform the erasing or programming after checking the FDERSTA bit or FDPRSTA bit is "0". The erasing or programming becomes invalid when the FDERSTA bit or FDPRSTA bit is "1".

(Section 25.3.1 "Notes on debugs for self-programming codes")
[ ] Do not perform real time execution (GO execution) when the breakpoints are set in the flash self-programming sequence (from writing to the flash acceptor to writing to the flash data register). If the real time execution is performed with break points set in the sequence, the flash memory may not be reprogrammed.
[ ] Do not perform STEP execution in the flash self-programming sequence (from writing to the flash acceptor to writing to the flash data register). If the STEP execution is performed in the sequence, the flash self-rewrite may not be reprogrammed.
[ ] The DTU8 may respond an error message "ICE error: 6601H Failed to control target LSI. Please check the connection." and after that, may not respond to any further debug commands. In that case, reset the DTU8, EASE1000 and the target system.

(Section 25.3.2 "Programming the program memory")
Figure 25-1 Flow chart for erasing the program memory
[ ] If any areas used for running the program were erased, the MCU would be uncontrollable by the software. Erase the areas unused for running the program.
[ ] The CPU is in the wait state for max.85ms during the block erase or sector erase. Make a proper WDT clear process, so that it does get overflow during the erase.
[ ] Interrupts are prohibited during the block erase or sector erase. Disable the interrupt by using __DI() function before setting FERS bit or FSERS bit.
[ ] Insert the following codes on the next code of setting the FERS bit or FSERS bit, otherwise the operation is not guranteed.
```
__asm("DW 0FE9FH");
__asm("NOP");
```
[ ] For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

Figure 25-2 Flow chart for programming the program memory
[ ] If any areas used for running the program were reprogrammed, the MCU would be uncontrollable by the software. Program the areas unused for running the program.
[ ] The CPU is in the wait state for max.80μs during the block erase or sector erase. Make a proper WDT clear process, so that it does get overflow during the erase.
[ ] Interrupts are prohibited during data flash write. Disable the interrupt by using __DI() function before setting write data to FLASHD1 register. Also, clear the WDT counter so that the WDT interrupt does not occur during the data set.
[ ] Insert the following codes on the next code of setting the FLASHD1 register, otherwise the operation is not guranteed.
```
__asm("DW 0FE9FH");
__asm("NOP");
```
[ ] For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

(Section 25.3.3 "Programming the data flash memory")
Figure 25-3 Flow chart for erasing the data flash memory
[ ] The CPU contines running the program code while erasing the data flash. Do not enter to STOP mode, STOP-D mode or HALT-H mode during the erase. Also, reset FSELF to "0" after the erase ended.
[ ] The data flash areas are not readable during the erase.
[ ] Interrupts are prohibited during the block erase or sector erase. Disable the interrupt by using __DI() function before setting FERS bit or FSERS bit.
[ ] Insert the following codes on the next code of setting the FERS bit or FSERS bit, otherwise the operation is not guranteed.
```
__asm("DW 0FE9FH");
__asm("NOP");
```
[ ] For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

Figure 25-4 Flow chart for programming the data flash memory
[ ] The CPU contines running the program code while programming the data flash. Do not enter to STOP mode ,STOP-D mode or HALT-H mode while programming the data flash. Also, reset FSELF to "0" after the programming ended.
[ ] The data flash areas are not readable while programming the data flash.
[ ] Interrupts are prohibited during data flash write. Disable the interrupt by using __DI() function before setting write data to FLASHD0L register. Also, clear the WDT counter so that the WDT interrupt does not occur during the data set.
[ ] Insert the following codes on the next code of setting the FLASHD0L register, otherwise the operation is not guranteed.
```
__asm("DW 0FE9FH");
__asm("NOP");
```
[ ] For enabling the interrupt, put the __EI () function immdiately after the "__asm("NOP");".

(Section 25.3.4 "Notes in Use of the self-programming")
[ ] Enable and select the high-speed clock for the system clock when performing the block/sector erase or programming the flash memory. For details on how to enable the high-speed oscillation and switch the system clock, see Chapter 6 "Clock Generation Circuit".
[ ] In case a power failure occurred or the operation was terminated forcibly during block/sector erase or programming the flash memory, the programmed data cannot be guranteed. Please retry erasing and programming the flash memory.
[ ] In case the power failure occurred or the operation was terminated forcibly during block/sector erase or programming the program memory area contains the address "0:0000H" and the MCU did not restart running, please reprogram a program code by using the on-chip debug emualtor (EASE1000) or using the ISP function.

(Section 25.4.3 "Communication command")
[ ] The units of programming the program memory is 4 bytes. Specify the lowest 4bit as 4 bytes boundary (0H, 4H, 8H and CH).

(Section 25.4.4 "ISP mode transition")
[ ] Complete the ISP mode transfer command (point B in the Figure 25-6) through the end of the Initial setting command (1) within 55ms.

Chapter 28 On-Chip Debug Function
(Section 28.3 "Notes on Debug with EASE1000")
[ ] Do not mount a component that makes the level fixed to H on the RESET_N pin.
[ ] Do not mount a component on the P00/TEST0 pin.
[ ] Do not write an application code that sets P00 to the output mode to the LSI. Since this code is executed before EASE1000 accesses the LSI, the P00/TEST0 pin is set to the output mode, so the LSI can no longer connect the EASE1000. Please note that the input/output mode of P00 cannot be initialized from EASE1000.
[ ] Validate the ROM code on user production board without LAPIS semiconductor development tool EASE1000.
[ ] Do not connect EASE1000 when measure the consumption current of a target system. The on-chip debugging circuit in target LSI influences, and consumption current increases.
[ ] When using the 3.3VOUT power supply of EASE1000, do not apply power of the user target system to the VDD pin of the target LSI. If both power supplies are connected, it may damage EASE1000 or cause an electric shock or fire.
[ ] Supply 3.0V to 5.5V to the V_{DD} pin when programming the Flash on the MCU by using the EASE1000.
[ ] Please do not apply LSIs being used for debugging to mass production.

(Section 28.4 "Overview of On-Chip Debug Function")
[ ] The ROM unused area access reset does not occur in the on-chip debug mode. The ROM unused area access break can be generated by setting the function on the DTU8 debugger.
[ ] The RAM parity error reset does not occur in the on-chip debug mode. The RAM parity error break can be generated on the DTU8 debugger.

Appendix A Register List (SFR List)
[ ] Confirm there are some SFRs that have undefined initial value.

Appendix B Package Dimensions
[ ] The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

Appendix D Application Circuit Example
[ ] Place the capacitor for V_{DDL} pin as close to the LSI power pins as possible.

Electrical Characteristics
See ML62Q1200A Data Sheet.
• External capacitor for power pins
[ ] CL = 1.0uF (for VDDL pin), [ ] CV = 1.0uF or larger (for VDD)
• Operating Voltage
[ ] +1.6V to +5.5V (30kHz to 4MHz), +1.8V to +5.5V (30kHz to 25MHz),
• Operating Temperature
[ ] -40°C to +105°C (Reading the Flash), -40°C to +105°C (Programming the Flash)

REVISION HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1200A-04	Nov 1, 2017	-	-	Formal the 4th Revision. (The 1st to 3rd Revision are skipped to match the revision number to Japanese User's Manual.)
FEUL62Q1200A-05	Feb 19, 2018	1-3	1-3	Added the restriction of operate-able CPU mode for the DMA controller.
		1-22	1-22	Change a word in the Note.
		2-4	2-4	Corrected the hex form of result value in the saturating multiply-accumulate operation.
		3-2	3-2	Corrected the I/O attribute of RESET_N pin.
		4-1	4-1	Deleted "Watch Dog Timer(WDT) operation in STOP mode and STOP-D mode is selectable by the code option"
		4-6~4-13	4-6~4-13	Added "The bits are unwriteable when the products do not have the peripheral function and always return "0" for reading."
		4-16	4-16	Corrected the waveform of LSCLK in the Figure 4-4.
		4-19	4-19	Added the CRC calculator and Multiplier/Divider in the Table 4-3.
		5-6~5-12, 5-14,5-16 ~5-21	5-6~5-12, 5-14,5-16 ~5-21	Changed the access type of reserve bits.
		6-1	6-1	Added "(VDD=1.8V or higher)" as the condition of ±1%.
		6-2	6-2	Changed the way of describing the WDT clock in the Figure 6-1.
		6-3	6-3	Deleted "Approx.1.024kHz" in the Table 6-2.
		6-10,6-13	6-10,6-13	Corrected the stabilization time "2.5ms" to "2ms"
		6-11	6-11	Added the description about the Low-Speed RC Oscillation Frequency Adjustment.
		6-14	6-14	Corrected the frequency of WDT dedicated RC oscillation clock "1.024kHz" to "1kHz". Deleted the description of that WDT clock is selectable by the Code option.
		7-1	7-1	Deleted unnecessary parts in the Figure 7-1.
		7-8	7-8	Added the [Note] on how to measure times using the Low-speed time base counter.
		8-1	8-1	Minor-changed the description in the Features. Minor-changed the Figure 8-1.
		8-2	,8-2	Minor-changed the Figure 8-2. Deleted a note "Select 16-bit timer mode when using the 16-bit timer trigger."
		8-3	8-3	Added "EXTRG0-1" to the List of Pins. Deleted the [Note].
		8-4,8-5	8-4,8-5	Corrected R/W attribution on the Register List.
		8-6,8-9	8-6,8-9	Added "(THnSTAT/THnHSTAT bits of TMHSTAT register are "0")" in the [Note].
		8-8	8-8	Added the note on the external trigger.
		8-13,8-15	8-13,8-15	Added "Set TMHSTR when the timer n is stopped (THnSTAT/THnHSTAT bits of TMHSTAT register are "0")" in the [Note].
		-	8-21	Added the 8.3.3. External Input Count Timing.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1200A-05	Feb 19, 2018	9-2,9-21~ 9-23,9-42	9-2,9-21~ 9-23,9-42	Changed the signal name COMP0INT to COMP0OUT.
		9-9	9-9	Added description about PWM2 mode.
		9-9,9-10	9-9,9-10	Added description about PWM1 mode.
		9-16	9-16	Added description about FTnSTPO bit.
		9-21	9-21	WDT dedicated clock CR1KHz → WDT dedicated clock RC1KHz
		9-22,9-23	9-22,9-23	Added "RC1KHz" as a even trigger source in the description for FTMn Trigger Register 1.
		9-31	9-31	Added the [Note] about setting the FTCSTR register.
		9-32	9-32	Added the [Note] about setting the FTCSTP register.
		9-38	9-38	Added the Figure 9-3.
		9-38~9-46	9-39~9-47	Shifted the Figure numbers.
		10-5	10-5	Added the note on the behavior of WDT overflow.
		11-8	11-8	Added a note on the operation to the SDnBUF register.
		11-11	11-11	Corrected description of SnEN bit, Un0EN bit and Un1EN bit. "start" → "enable"
		11-13	11-13	Added a note on the frequency of transfer clock.
		11-21	11-21	Corrected description about Un0FER bit.
		11-22	11-22	Corrected description about Un0PER bit.
		11-23	11-23	Corrected description about Un1FER bit.
		11-24	11-24	Corrected description about Un1PER bit.
		11-22,11-24	11-22,11-24	Added a note on how to write Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit.
		11-25~11-29	11-25~11-29	Changed the waveform of SIUn0INT in the Figure 11-2~11-10.
		11-31	11-31	Added the Table 11-3.
		11-32	11-32	Added the Figure 11-11.
		11-34	11-34	Corrected the Actual Baud Rate in the Table 11-4.
		11-39	11-39	Added the Table 11-5 and 11-6.
		12-1	12-1	Changed the description in the 12.1.1 Features.
		12-4	12-4	Added notes on the usage of I2C bus.
		12-9	12-9	Added notes when writing the I2UM0CON register.
		12-11, 12-26, 12-27	12-11, 12-26, 12-27	Added a note on the I2C operation clock.
		12-12	12-12	Changed the description on the note.
		12-16	12-16	Added the note.
		12-17, 12-25	12-17, 12-25	Deleted the note about the frequency of system clock.
		12-19	12-19	Changed the description on the note.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1200A-05	Feb 19, 2018	13-1	13-1	Added a description in the 13.1.1 Features.
		13-7	13-7	Changed the description on the note and added a note.
		13-9, 13-14, 13-15	13-9, 13-14, 13-15	Added a note on the I2C operation clock.
		13-11	13-11	Corrected a typo in the description of 13.3.1.1 (I2C bus → I2C master).
		14-1	14-1	Added the description of operation mode.
		14-7,14-8	14-7,14-8	Added a description about the target address in the 16-bit data transfer.
		14-12	14-12	Added description about transfer interval time and added a note about the CPU mode in the [Note].
		14-13	14-13	Changed "UA00BRC = 0x01" to "UA00BRC = 0x02" in the flow chart.
		-	14-14	Added the 14.3.3 SSIO Continuous Transmission.
		-	14-15	Added 14.3.4. DMA Transfer Target Block
		-	17-24	Added 17.3.8 Notes for using the P00/TEST0 pin
		19-1, A-4	19-1,A-4	Corrected the names of registers related to CRC calculation.
		20-3	20-3	Added the note for preventing the influence of noise.
		21-3	21-3	Added the note for preventing the influence of noise.
		23-1	23-1	Added "Voltage input from the V_{DD} pin" as one of selectable reference voltage.
		23-3	23-3	Deleted a note and added a note.
		23-8	23-8	Added description about SAULS07~SAULS00.
		-	23-9	Added the notes on the SADULS0 register.
		23-9	23-10	Added description about SAULS16 bit and the notes.
		23-11	23-12	Added a note about setting SARUN bit.
		23-15	23-16	Added a description about SALEN bit.
		23-23	23-24	Changed the timing of setting the SARUN bit and starting the A/D conversion in the Figure 23-3 and 23-4.
		-	23-25	Added the Figure 23-5.
		25-5,25-6	25-5,25-6	Added a note about access to unused area.
		25-9	25-9	Added a note about FSELF bit.
		25-17	25-17	Changed the min. baudrate from 2400bps to 4800bps.
		25-19	25-19	Changed the Figure 25-6 and added a note.
		26-4	26-4	Changed the description about the example of code option data definition.
		28-2	28-2	Added two notes on debug with EASE1000.
		A1~A14	A1~A14	Deleted "R", "8" and "0x00" in the Reserved registers.
FEUL62Q1200A-06	Feb 8, 2019	1-4	1-4	[1.1] Added VDD as selection of the A/D converter referece voltage.
		1-19	1-19	[1.3.2] Deleted "(w/ Pull-UP)" in the description for RESET_N pin.
		3-1	3-1	Added "(when ELEVEL of PSW = 2 or larger)" to the description about the CPU reset.
		3-3	3-3	[3.2.2] Changed description of RSTAT on the top.
		3-5	3-5	[3.2.3] Changed description of SRSTAT on the top.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1200A-06	Feb 8, 2019	4-19	4-20	[4.3.7] Added a note in the [Note].
		7-1	7-1	Change descriptions in 1.1 Features.
		7-6	7-6	[7.2.3] Added two notes in the [Note].
		8-15	8-15	Corrected the 2nd note in the [Note].
		9-10	9-10	[9.2.3] Corrected the value of FtnEA15-0 in the PWM2 mode.
		9-22	9-22	[9.2.10] Changed the description at the FTnSTS* bits is 01100.
		9-23	9-23	[9.2.11] Deleted RC1KHz as the event trigger source.
		9-24	9-24	[9.2.11] Corrected "HSCLK" to "Timer Clock" in the description of FTnTRF2-0 bits.
		9-37	9-37	[9.3.2.1] Added some description.
		10-5,-10	10-5,-10	Corrected "1kHz" to "1.024kHz".
		11-21	11-21	[11.2.14] Corrected description about Un0OER bit.
		11-34	11-34	[11.3.2.2] Added description about the base clock and the baud rate. Corrected the mathematical expression of the baud rate in the example.
		12-10	12-10	[12.2.7] Corrected the description of I2UM0MD1 to I2UM0MD0 bits.
		12-16	12-16	[12.2.12] Deleted a note in the [Note].
		13-1	13-1	[13.1] Added a note about the multi-master.
		14-7~9	14-7~9	[14.2.4] to [14.2.6] Corrected "DCnTN" to each symbol in the [Note].
		22-1	22-1	[22.1.1] Deleted "VLS0 reset/interrupt is disabled during on-chip debug".
		22-10	22-10	[22.3.1] Deleted Figure 22-2 " Operation Timing Diagram (when the VLS0 reset without sampling is selected)" and the related description, becuase the sampling is required in the super visoer mode.
		22-11~16	22-11~15	Shifted up the figure numbers.
		23-10	23-10	[23.2.4] Corrected the first note in the [Note]. Deleted "reading or".
		23-15	23-15	[23.2.10] Changed the description of A/D conversion interval time.
		24-1	24-1	Changed "CV=1μF" to "CV=1μF or larger" in the Figure 24-1.
		24-2	24-2	Added section 24.2.1 "Reference Voltage Output".
		25-18~24	25-18~24	[25.4.3 to 25.4.5.4] Overall modification of tables , figures and descriptions.
		28-1	28-1	Changed "CV=1μF" to "CV=1μF or larger" in the Figure 28-1 and 28-2.
		28-3	28-3	[28.4.1] Changed the description and added the table.
		29-8~11	29-8~12	Changed the structure of section 29.2.7 ~ 27.2.11 by moving to section 29.3 and updated the contents.
		29-10	29-10~11	[29.3.3] Updated the contents.
		D-1	D-1	Added a pull-down resiter at the EXI0 pin. Changed the capacitance value of Cv to "1.0uF or larger".